

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI  
Corso di Laurea Triennale in Informatica

# SISTEMI ADATTIVI SEMANTICI

Tesi di Laurea in Intelligenza Artificiale: Rappresentazione della  
Conoscenza

Relatore:  
Chiar.mo Prof.  
MAURO GASPARI

Presentata da:  
DANIELE BELLINI

Sessione 2 Anno Accademico 2010-2011

# Indice

<b>Introduzione</b>	<b>3</b>
<b>1 Classificazione dei SASs</b>	<b>5</b>
1.1 Semantic Web . . . . .	5
1.2 Web 2.0 . . . . .	6
1.3 SASs forti . . . . .	8
1.4 SASs deboli . . . . .	9
<b>2 Principali strumenti e tecnologie utilizzate dai SASs</b>	<b>10</b>
2.1 Tecnologie proposte per semantica forte . . . . .	10
2.1.1 Linguaggi ontologici (usati per rappresentare le ontologie) . . . .	10
2.1.2 Linguaggi e ontologie che descrivono servizi del Semantic Web . .	14
2.1.3 Linguaggi che definiscono regole di markup per linguaggi ontologici	16
2.1.4 Linguaggi e tools per interrogare e ragionare su ontologie . . . . .	16
2.2 Tecnologie proposte per semantica debole e approcci ibridi . . . . .	17
<b>3 Funzionamento dei SASs</b>	<b>20</b>
3.1 Acquisizione e scambio dei dati tra SASs . . . . .	20
3.1.1 Rappresentazioni utilizzate per i tag . . . . .	21
3.1.2 Tecniche per gestire lo scambio di tag tra SASs . . . . .	23
3.2 Gestione della conoscenza . . . . .	26
3.2.1 Modellazione del dominio, del contesto e dell'utente . . . . .	27
3.2.2 Estensione della conoscenza dei modelli di dominio, contesto e utente	29
3.3 Processi di adattamento . . . . .	38
3.3.1 Adattamento dei contenuti e loro suggerimento . . . . .	38
3.3.2 Adattamento della struttura e della presentazione . . . . .	42
3.3.3 Adattamento per supportare le annotazioni . . . . .	45
<b>4 “Pillo SAS’s”: la realizzazione di un SAS</b>	<b>49</b>
4.1 Acquisizione . . . . .	50
4.2 Gestione della conoscenza . . . . .	51

4.3	Processo d'adattamento . . . . .	52
	<b>Conclusioni</b>	<b>57</b>
	Bibliografia . . . . .	60

# Introduzione

Lo scopo principale di questa tesi è studiare il funzionamento e le tecnologie utilizzate dai “sistemi adattivi semantici” (SASs: Semantic Adaptive Systems).

Come prima cosa è bene definire cos'è un sistema adattivo.

*“Si definiscono adattivi (adaptive) i sistemi che adeguano in modo automatico il funzionamento dell'applicazione alle caratteristiche del contesto e dell'utente. Più precisamente si definisce adattivo un sistema che adatta il contenuto, la struttura e le modalità di presentazione alle caratteristiche dell'utente (user data), al comportamento d'utilizzo (usage data), all'ambiente d'utilizzo (environment data), ed è in grado di raccogliere ed integrare le informazioni relative a queste caratteristiche all'interno di un modello del contesto, e di applicare poi tale modello per adattare alcuni aspetti del sistema (come ad esempio l'interfaccia, la struttura di navigazione e la qualità-quantità dell'informazione inviata)”<sup>1</sup>.*

Esistono diverse tipologie di sistemi adattivi e in questa tesi sono presentati in modo approfondito i sistemi adattivi semantici, ovvero sistemi che utilizzano tecniche semantiche per compiere il processo di adattamento, e per modellare il dominio, il contesto e le caratteristiche dell'utente. Le tecniche semantiche sono per l'appunto tecniche che si basano e sfruttano informazioni (metadati) attribuite a varie risorse che gli conferiscono un significato semantico, adatto alla comprensione da parte di una macchina al fine di consentirne l'elaborazione e l'estrazione di determinati tipi di conoscenza. Alcuni esempi di queste tecniche sono il recupero di risorse, il filtering di contenuti e informazioni, l'applicazione di regole d'inferenza su dati semantici, e molte altre. Inizialmente sarà mostrata una classificazione dei SASs (d'ora in poi verrà utilizzata questa notazione abbreviata per indicare “Sistemi Adattivi Semantici” o “Semantic Adaptive Systems”), e delle tecnologie utilizzate da essi al fine di chiarire i concetti base di questi sistemi. Una volta introdotta

---

<sup>1</sup>(definizione adattata da: <http://mobile.html.it>)

te queste prime nozioni, si passerà a mostrare i dettagli delle varie tecniche e modalità di funzionamento dei SASs facendo anche riferimento ad esempi specifici di alcuni di questi sistemi. Infine sarà illustrato il lavoro svolto nella realizzazione e implementazione di un SAS, il sistema adattivo “Pillo SAS’s”, sviluppato a fini didattici dall’autore di questa tesi per comprendere meglio, tramite un esempio, il funzionamento di tali sistemi, e sperimentare l’utilizzo delle tecnologie del Semantic Web nei sistemi adattivi.

# Capitolo 1

## Classificazione dei SASs

I SASs si possono suddividere principalmente in due categorie: “SASs forti” e “SASs deboli”. I SASs forti utilizzano “tecniche semantiche forti”, ovvero tecniche basate sul Semantic Web e rappresentazioni di conoscenza esplicita, mentre i SASs deboli utilizzano “tecniche semantiche deboli”, ovvero tecniche più vicine al Web 2.0, social annotation e approcci ibridi.

Esempi di tecniche semantiche forti sono ad esempio quelle mirate al recupero di risorse, allo scambio e all’integrazione di informazioni, o all’applicazione di regole di ragionamento per estrarre conoscenze implicite da quelle esplicite.

Esempi di tecniche semantiche deboli sono quelle di estrazione e inferenza di conoscenza da azioni di “tagging” o da tag generati e posti su risorse da utenti diversi in ambienti sociali come ad esempio i più odierni social network.

Per quanto riguarda invece gli approcci ibridi, essi, anche se classificati come tecniche di semantica debole fanno uso di approcci misti, ovvero mescolano ed utilizzano assieme tecniche semantiche forti con quelle deboli.

### 1.1 Semantic Web

Lo scopo principale del Semantic Web è di fornire un framework comune per rappresentare documenti, risorse e informazioni nel Web, al fine di garantirne una facile e agile condivisione, scambio e riutilizzo delle informazioni da parte delle applicazioni Web. In altre parole si vuole raggiungere l’obiettivo di rendere l’informazione automatica comprensibile e utilizzabile da agenti software, integrare contenuti di risorse differenti, ottimizzare le ricerche e il recupero delle risorse nel Web, e garantire maggior fiducia rispetto ai contenuti presenti al suo interno. I documenti e le risorse del Semantic Web

sono descritti mediante l'uso di linguaggi formali, modellano dunque domini di conoscenza formali (stabili), ed esprimono concetti e riferimenti mediante l'uso di ontologie<sup>1</sup>. L'attribuzione di un significato semantico alle risorse nel Web, è fatta generalmente dal loro creatore, il quale attribuisce un significato alle informazioni mappandole con concetti espressi in una qualche ontologia di utilizzo. Tali informazioni possono poi essere utilizzate da agenti software intelligenti, che processano, riconoscono, e possono ragionare sulle ontologie: è possibile definire vari tipi di regole che permettono di estendere il dominio di un'ontologia, oppure che fanno delle inferenze su esse, o che specificano determinate policy di sicurezza, o che generano determinati cambiamenti o azioni rispetto alle ontologie prese in esame.

## 1.2 Web 2.0

Web 2.0 si potrebbe definire come “il Web di ultima generazione”, è una sorta di nuova visione del Web, in cui prevale rispetto al vecchio Web (1.0) l'interazione degli utenti, e l'aspetto sociale che si viene a creare tra essi. Chat, blog, social network, e-commerce, e molte altre applicazioni permettono di ottenere un alto livello d'interattività degli utenti, che condividono, generano e si scambiano tra loro informazioni sul Web. Nell'ambito particolare di questo studio si prende in esame l'utilizzo dei tag e delle social annotation che gli utenti pongono su determinate risorse e documenti, analizzando come i sistemi adattivi li possano utilizzare nel processo di adattamento. In particolare quindi sono d'interesse Weblogs, Wiki tools e Collaborative Tagging Systems, cioè sistemi in cui gli utenti possono condividere e caricare risorse (documenti di testo, foto, e altri media multimediali) sul Web, e altri utenti possono commentarle e valutarle. In tali sistemi la conoscenza e le informazioni sono rappresentate per mezzo di “folksonomie”<sup>2</sup>, ovvero una sorta

---

<sup>1</sup>Ontologia: non è stata ancora trovata una definizione specifica universalmente accettata di ontologia; alcune tra le definizioni più note sono:

- rappresentazione formale, condivisa ed esplicita di una concettualizzazione di un dominio di interesse (definizione di Tom Gruber);
- un vocabolario può essere considerato come una speciale forma di ontologia, o più semplicemente come una collezione di URI con un significato descritto (di solito in modo informale) (definizione di <http://semanticWeb.org>);

<sup>2</sup>Folksonomia:

- è un neologismo derivato dal termine di lingua inglese folksonomy che descrive una categorizzazione di informazioni generata dagli utenti mediante l'utilizzo di parole chiave (o tag) scelte liberamente. Il termine è formato dall'unione di due parole, “folk” e “tassonomia”; una folksonomia è, pertanto,

di sistemi di classificazione dei tag, che non hanno una struttura prefissata, e sono quindi in grado di modellare anche domini di conoscenza informali, molto instabili, che possono cambiare rapidamente nel tempo. Inoltre, con queste tecniche di “tagging” del Web 2.0, è possibile sia identificare un aspetto comune e prevalente che emerge dai tag degli utenti su una determinata risorsa, sia punti di vista diversi, ovvero aspetti più sfaccettati che quella risorsa ha nello specifico per ogni utente.

Riassumendo schematicamente quanto visto su Semantic Web e Web 2.0, possiamo elencare le differenze fondamentali tra essi:

Semantic Web:

- annotazioni definite da chi crea la risorsa;
- utilizzo di ontologie;
- modellazione di domini formali (stabili);
- agevole per lo scambio, riuso e integrazione delle informazioni;

Web 2.0:

- annotazioni poste dagli utenti che visualizzano la risorsa;
- utilizzo di folksonomie;
- modellazione di domini informali (instabili);
- mostra sia aspetti prevalenti, che una sfaccettatura di aspetti specifici;

Problematiche di Semantic Web:

Le tecnologie basate su Semantic Web possono presentare tre problematiche principali:

- alti costi e difficoltà di sviluppo;

---

una tassonomia creata da chi la usa, in base a criteri individuali. In maniera più semplice e concreta, questo termine si riferisce alla metodologia utilizzata da gruppi di persone che collaborano spontaneamente per organizzare in categorie le informazioni disponibili attraverso internet (Web 2.0). Questo fenomeno, in contrasto con i metodi di classificazione formale (in particolare con la tassonomia classica), cresce soprattutto in comunità non gerarchiche legate ad applicazioni Web, attraverso le quali vengono diffusi contenuti testuali e/o multimediali (definizione di Wikipedia).

- sono il risultato di una libera e personale azione di “tagging” di informazioni e oggetti al fine di poterli facilmente e agevolmente recuperare. Il tagging è fatto in un ambiente sociale, generalmente aperto e condiviso con altri utenti. Le folksonomie sono generate dall’azione di taggare da parte degli utenti che andranno ad utilizzare quelle informazioni. I valori assunti da questi tag sono definiti dagli utenti con il loro proprio vocabolario, e con significati propri ed espliciti. (definizione di Ilaria Torre)



- difficoltà nell'allineamento e matching di ontologie di sistemi differenti;
- difficoltà nel trovare un trade-off di usabilità e riusabilità nella definizione di concetti per mezzo di ontologie, che saranno usati come annotazioni per le risorse.

Problematiche di Web 2.0:

- difficoltà nel classificare i tag;
- errori d'interpretazione dei tag;

Si può inoltre fare un'ulteriore classificazione relativamente al Semantic Web, vista come una sorta di parallelismo con l'IA debole/forte definite da Searle<sup>3</sup>:

#### **Semantic Web debole:**

Si pone l'obiettivo di accrescere i significati attribuiti alle risorse, al fine di migliorare i servizi forniti all'utente, indipendentemente dal tipo di annotazioni semantiche utilizzate. Si presenta meno estendibile e potente del Semantic Web forte, ma è più efficace e applicabile a breve termine.

#### **Semantic Web forte:**

Si pone l'obiettivo di far comprendere i dati alle macchine che li utilizzano, sempre con la finalità di migliorare i servizi verso gli utenti. È più potente di Semantic Web debole, e non è limitato nella rappresentazione della conoscenza, tuttavia presenta una maggiore complessità e maggiori costi nello sviluppo.

Introdotti brevemente i concetti di Semantic Web e Web 2.0, e le relative distinzioni tra Semantic Web forte/debole, si può definire una classificazione di SASs forti/deboli.

## **1.3 SASs forti**

I SASs forti sono sistemi che fanno uso di tecniche semantiche forti, e dunque più vicine alle tecnologie del Semantic Web e più in particolare al Semantic Web forte; quindi fanno uso di ontologie per rappresentare e modellare domini

---

<sup>3</sup>Searle: nella distinzione tra Semantic Web forte/debole si può fare un parallelismo con l'IA forte/debole di Searle (famoso filosofo statunitense) che definisce programmi che utilizzano l'IA debole, usati per studiare o risolvere specifici problemi che non possono essere compresi pienamente nei limiti della capacità umana (come ad esempio le mosse vincenti del gioco degli scacchi); mentre per IA forte definisce l'idea di una macchina pensante (sapiente e cosciente), che sia in grado veramente di ragionare e risolvere problemi di ogni tipo.

di conoscenza, in particolare domini formali e stabili, e sviluppano il processo di adattamento con tecniche di ragionamento su tali ontologie (molto spesso utilizzano linguaggi e tool specifici per costruire e interrogare ontologie). Si presentano da un lato come sistemi molto potenti nella rappresentazione della conoscenza, ma allo stesso tempo anche costosi e complessi nello sviluppo.

## 1.4 SASs deboli

I SASs deboli invece sono sistemi che fanno uso di tecniche semantiche deboli, e quindi Semantic Web debole e social annotation di Web 2.0. Sono meno potenti nella rappresentazione della conoscenza, ma attraverso l'uso di folksonomie permettono di modellare domini instabili. Sono stati proposti, nello sviluppo dei SASs, anche approcci ibridi, ovvero approcci misti che cercano di utilizzare assieme, sia tecniche del Semantic Web, che del Web 2.0: ad esempio è stato proposto di applicare le tecnologie del Semantic Web a dati del Social Web, oppure altri approcci misti hanno proposto di incorporare il contesto sociale (del Social Web) all'interno delle rappresentazioni delle ontologie. Tali approcci ibridi sono stati attribuiti nella classificazione ai SASs deboli.

## Capitolo 2

# Principali strumenti e tecnologie utilizzate dai SASs

Dopo aver mostrato che i SASs utilizzano tecnologie di Web 2.0 e Semantic Web, è possibile analizzare in modo più dettagliato i tipi di strumenti e tecnologie utilizzate per la loro realizzazione.

### 2.1 Tecnologie proposte per semantica forte

Qui di seguito saranno mostrati brevemente i principali strumenti per la rappresentazione di risorse RDF[28] e di ontologie come RDFS[30], OWL[32] e un breve accenno su F-logic[39]; altri strumenti per la rappresentazione di diversi tipi di conoscenza, come ad esempio Dublin Core[36] per modellare informazioni su risorse Web, o FOAF[34], utile per modellare le caratteristiche di un utente; e infine alcuni strumenti per rappresentare regole dei linguaggi ontologici, per interrogare e ragionare su ontologie.

#### 2.1.1 Linguaggi ontologici (usati per rappresentare le ontologie)

##### **RDF per rappresentare risorse**

Il “Resource Description Framework” (RDF) è uno strumento proposto dal W3C per rappresentare informazioni sul Web. RDF dunque è un metalinguaggio che descrive delle risorse (che non necessariamente devono stare sul Web). RDF presenta due componenti principali: RDF Model and Syntax[29],

che definisce il modello dei dati RDF e la sintassi utilizzata per specificare tale modello (spesso è utilizzata la sintassi XML); e RDF Schema (RDFS)[30] che consente di definire il significato e le caratteristiche delle proprietà relative alle risorse descritte nel modello.

### **RDF Model and Syntax:**

Il modello RDF è formato da risorse, proprietà e valori:

- ogni risorsa descritta da RDF è identificata da un URI (Universal Resource Identifier), che gli consente una identificazione univoca;
- le proprietà sono relazioni che legano tra loro risorse e valori, e sono anch'esse identificate da URI;
- i valori invece sono tipi di dato primitivo (stringhe), ma possono anche contenere una stringa con un URI di una qualche risorsa;

L'unità base di rappresentazione di un'informazione in RDF è lo statement, che è rappresentato in forma di tripla "soggetto-predicato-oggetto", dove il soggetto è una risorsa, il predicato è una proprietà, e l'oggetto è un valore (e quindi può essere anche un URI che punta ad un'altra risorsa). Quindi con RDF è possibile definire un modello che descriva le relazioni tra varie risorse, basandosi su opportune proprietà (anch'esse come le risorse identificate da un nome univoco).

### **RDF Schema:**

RDF Schema è un modello concettuale che permette di descrivere la struttura e l'organizzazione degli statement RDF. Ad esempio, RDF Schema permette di dichiarare proprietà e definire le relazioni che legano tali proprietà alle risorse. Inoltre con RDF Schema è possibile definire nuovi tipi di classi e sottoclassi di proprietà (e quindi gerarchie di classi), e definire nuovi vocabolari e schemi per i metadati RDF. I vari vocabolari definiti da RDF Schema possono essere inoltre usati assieme per modellare dunque una vasta gamma di domini differenti. RDF Schema rappresenta quindi le classi di ontologie e le proprietà, mentre RDF ne rappresenta le istanze. Esempi di alcuni schemi e vocabolari RDF sono:

- FOAF(Friend of a Friend): uno schema per descrivere persone, le loro attività e relazioni con altre persone e oggetti; molto utile per modellare profili di utenti;

- Dublin Core: uno schema inizialmente nato per descrivere le risorse delle digital library, in seguito è diventato uno schema per descrivere metadati (quindi i termini e le relazioni tra essi), tanto da diventare uno standard ISO;
- SKOS (Simple Knowledge Organisation Systems)[38]: è un modello di dati per condividere e collegare le conoscenze organizzate di vari sistemi, come ad esempio thesauri e tassonomie; tale modello fornisce anche un percorso di migrazione di tali conoscenze organizzate sul Semantic Web;
- RSS (RDF Site Summary 1.0): schema che descrive la struttura del formato RSS per la distribuzione di contenuti sul Web. Tale formato è utilizzato principalmente per la descrizione di metadati riguardo varie risorse Web;
- Music Vocabulary: un vocabolario che descrive i capolavori, spettacoli e concerti di musica classica; lo schema definisce classi per opere musicali e interpreti e le relative proprietà;
- Beer Ontology: un'ontologia che modella birrai e tipi di birra.

## OWL

“Ontology Web Language” è per l'appunto un linguaggio per definire e istanziare ontologie. È una recommendation del W3C, e si presenta come una tecnologia potenzialmente più espressiva ed estesa rispetto a RDF.

OWL è un linguaggio ontologico che nasce per superare alcune problematiche e limitazioni di RDF e RDFS. Il problema principale deriva dalla logica predicativa del prim'ordine, che si presenta estremamente complessa, e RDF è in grado di esprimerne solo una porzione molto ristretta. Inoltre questa logica, se presa complessivamente, non è nemmeno computabile, mentre lo possono essere delle logiche costituite da sottoinsiemi di suoi operatori. Tali sottoinsiemi sono studiati dalle logiche descrittive, o “description logics”, che sono state adottate per la formulazione di un nuovo standard, più ricco ed espressivo di RDF, per l'appunto l'OWL. Già alcuni linguaggi precedenti e contemporanei a RDF, come DAML e OIL, avevano iniziato a trattare una serie di costrutti per le description logics; successivamente sono confluiti in DAML+OIL e da questa base il W3C è partito a sviluppare OWL. RDF si presenta dunque da un lato come uno strumento anche fin troppo potente, tanto che la sua totale estendibilità nella rappresentazione di concetti e conoscenza può portare a delle difficoltà computazionali. Dall'altro lato invece gli mancano determinati costrutti (come ad esempio predicare la cardinalità,

l'uguaglianza, la simmetria ecc.). Per tali motivi RDF è risultato uno strumento intrattabile e inutilizzabile in alcuni casi, e OWL è stato creato proprio per risolvere tali problemi e per introdurre le description logics all'interno del linguaggio.

L'OWL offre nuovi costrutti e vocabolari per descrivere proprietà e classi: relazioni tra classi (es. predicare la disgiunzione), cardinalità (es. "esattamente uno"), uguaglianza, una maggiore tipizzazione delle proprietà e delle loro caratteristiche (es. "simmetria", "transitività"), classi enumerative, e molte altre. Il linguaggio OWL prevede tre tipi di sottolinguaggi secondo un livello progressivamente più elevato di espressività:

- OWL Lite: linguaggio che permette la creazione di classificazioni in gerarchie, e di esprimere semplici vincoli. Utile per la generazione di thesauri e tassonomie;
- OWL DL: è un linguaggio che garantisce la massima espressività senza perdere in completezza computazionale (è garantito che tutte le implicazioni saranno computate) e in decidibilità (tutte le computazioni saranno completate in un tempo finito) nei "sistemi di ragionamento". L'OWL DL include tutti i costrutti del linguaggio al pari delle loro restrizioni, quale la separazione dei tipi: una risorsa dichiarata come classe non può essere allo stesso tempo un individual o una proprietà, e una proprietà non può essere contemporaneamente anche una classe o un individual. L'OWL DL è così denominato per la sua affinità con le description logics, un campo di ricerca che studia una particolare porzione decisionale della logica di base atta a modellare conoscenza terminologica e quindi ontologie. OWL DL è stato progettato proprio per supportare il settore delle description logic esistente, e possiede specifiche proprietà computazionali per i "sistemi di ragionamento";
- OWL Full: linguaggio che permette la massima espressività e la "libertà sintattica" di RDF, pur rinunciando alle garanzie computazionali. Ad esempio, in OWL Full una classe può essere trattata simultaneamente come una collezione di individual e come un singolo. OWL Full consente a un'ontologia di accrescere il significato del vocabolario predefinito (RDF o OWL) a disposizione.

Quindi OWL Lite e OWL DL si possono vedere come una sorta di estensione ristretta di RDF e RDF Schema, con garanzie computazionali, mentre OWL Full rappresenta una vasta estensione di essi, ma senza garanzie computazionali. Tali linguaggi quindi non sono del tutto compatibili tra loro; inoltre ogni documento OWL è anche un documento RDF, ma non il vice versa.

## **F-logic**

“Frame logic” è un linguaggio ontologico e per la rappresentazione della conoscenza. Frame logic combina assieme i vantaggi della modellazione concettuale con quelli della modellazione object-oriented, e offre una sintassi dichiarativa, semplice e compatta, come pure una semantica e una base logica ben definite. Tra le altre caratteristiche di F-logic vi sono tutte quelle tipiche dei linguaggi object-oriented, quindi si possono definire identità di oggetti, oggetti complessi, ereditarietà, polimorfismo, incapsulamento, metodi d’interrogazione, ecc. Originariamente questo linguaggio era stato sviluppato per database deduttivi, in seguito però è stato preso in considerazione, e utilizzato per applicazioni del Semantic Web. F-logic è dunque considerato uno dei formalismi per la rappresentazione di ontologie, tuttavia strumenti come la description logic (OWL DL) sono divenuti più popolari e utilizzati rispetto a esso, infatti, le tecnologie maggiormente utilizzate dai SASs sono RDF e OWL, ma anche F-logic merita un breve accenno in questo lavoro.

Da queste ontologie, rappresentate con i linguaggi ontologici RDF e OWL, esistono poi vari strumenti e agenti software in grado di lavorare con esse, utilizzandole per interpretarle e costruirvi sopra dei ragionamenti.

## **Altri strumenti offerti dal Semantic Web**

Tra gli altri strumenti offerti dal Semantic Web, saranno brevemente presi in analisi linguaggi e ontologie che descrivono servizi del Semantic Web, linguaggi che definiscono regole di markup per linguaggi ontologici, linguaggi e tools per interrogare e ragionare su ontologie.

### **2.1.2 Linguaggi e ontologie che descrivono servizi del Semantic Web**

WSDL (Web Service Description Language) è un linguaggio basato su XML che descrive un modello per i servizi Web. Un servizio Web (o Web Service) è un sistema software progettato per supportare l’interoperabilità tra diverse applicazioni software che girano su varie piattaforme e utilizzano framework diversi. I Web Services sono caratterizzati dalla loro grande interoperabilità ed estendibilità, dovute al fatto di essere descritti con XML, dunque si presentano processabili ed elaborabili dalle macchine (WSDL è proprio uno dei linguaggi usati per descriverli). L’interfaccia fornita dai Web Services consente inoltre ad altri sistemi di attivare le funzionalità (o servizi) messi a

disposizione dal Web Service tramite semplici scambi di messaggi col servizio stesso (messaggi in formato XML scambiati con protocollo HTTP). Per introdurre il supporto all'uso di ontologie nei servizi Web è stata creata un'estensione di WSDL, SAWSDL (Semantic Annotation for WSDL and XML Schema)[40], che introduce un nuovo set di attributi per riferirsi al modello semantico. Introdotto il supporto al Semantic Web, serve ora una descrizione in modo semantico di tali servizi, e a questo si pone lo scopo di OWL-S[41], un'ontologia OWL che fornisce un linguaggio di markup per descrivere le proprietà e le capacità di un servizio Web. L'ontologia è suddivisa in tre parti:

- Service profile: descrive cosa fa il servizio; tra le informazioni che include, vi è il nome del servizio, una sua descrizione, le sue limitazioni sull'applicabilità, autore e informazioni sui contatti;
- Process model: descrive come un cliente può interagire con il servizio; la descrizione include una serie di input e output, pre-condizioni e il risultato dell'esecuzione del servizio su tali dati;
- Service grounding: specifica i dettagli di cui il cliente ha bisogno per interagire con il servizio, quindi il protocollo di comunicazione, il formato dei messaggi, numero delle porte, ecc.

Un'altra ontologia interessante da considerare nel campo dei servizi Web è WSMO (Web Service Modeling Ontology)[42], che fornisce un supporto per il deployment e per l'interoperatività tra servizi Web semantici. WSMO è costituito da quattro componenti:

- Obiettivi: descrivono gli obiettivi del cliente, quando consulta il servizio Web;
- Ontologie: formali descrizioni semantiche delle informazioni, utilizzate da tutte le altre componenti;
- Mediatori: collegamenti tra le varie componenti, che aiutano nella mediazione; provvedono all'interoperabilità tra ontologie differenti;
- Servizi Web: descrizioni semantiche dei servizi Web(descritti da varie ontologie, come OWL-S).

Le ontologie descritte con WSMO infine possono essere utilizzate da un altro linguaggio, WSLM (Web Service Modeling Language)[43], che provvede a rendere formale la sintassi e la semantica degli elementi WSMO, utilizzando diversi formalismi logici come logica del prim'ordine, description logic e logic programming.



### 2.1.3 Linguaggi che definiscono regole di markup per linguaggi ontologici

Correlati ai linguaggi ontologici, sviluppati per rappresentare ontologie, vi sono i linguaggi di markup, con i quali si esprimono le regole di riscrittura dei documenti e delle informazioni, attraverso opportuni marcatori. La principale famiglia di linguaggi di markup conosciuta è RuleML (Rule Markup Language)[44], ovvero un insieme di linguaggi per esprimere vari tipi di regole in XML. Più in particolare, nell'interesse del campo ontologico, è RDF RuleML, ovvero una versione rilasciata recentemente, che riguarda appunto regole di markup per ontologie RDF. Un altro linguaggio di markup, proposta del W3C, è SWRL (Semantic Web Rule Language)[45], ovvero un linguaggio che combina assieme i sottolinguaggi di OWL (OWL Lite e OWL DL) con Rule ML. SWRL si presenta compatibile sia con RDF sia con OWL, poiché identifica oggetti e risorse con URIs, e mostra un alto potere espressivo. Infine vi sono altri due linguaggi di markup sviluppati principalmente per l'interscambio di regole tra sistemi differenti, che sono R2ML[46] e RIF[47]. R2ML, sviluppato dal gruppo REWERSE, integra Rule ML e SWRL per garantire lo scambio di regole tra sistemi differenti, che usano questi due tipi di linguaggi di markup; RIF (Rule Interchange Format) è invece una proposta del W3C, per l'interscambio di regole tra sistemi che utilizzano linguaggi di markup diversi. RIF permette di interoperare con RDF e OWL.

### 2.1.4 Linguaggi e tools per interrogare e ragionare su ontologie

Dopo aver presentato una serie di linguaggi per definire ontologie e linguaggi per definire le regole di markup per linguaggi ontologici, si va ora a ispezionare una serie di linguaggi e tools utili per lavorare e manipolare tali ontologie. Più in particolare saranno mostrati alcuni esempi di linguaggi d'interrogazione (query) per le ontologie, utili per estrarre determinate informazioni e concetti da esse; e tools che sono in grado di operare, ragionare (fare inferenze, deduzioni, estrarre informazioni) sulle ontologie.

Il più usato linguaggio d'interrogazione per RDF è sicuramente SPARQL[48], accettato come Recommendation del W3C. Se le triple RDF che sono interrogate hanno elementi in comune, emerge da esse un "grafo della conoscenza", relativo a tali ontologie; SPARQL non fa altro che ricercare dei sotto-grafi corrispondenti alla richiesta effettuata dalla query. Grazie inoltre all'identificazione delle risorse in maniera univoca attraverso URIs di RDF, le query

esprimibili con SPARQL hanno la caratteristica di poter essere ben definite e non ambigue. Altri linguaggi d'interrogazione sono forniti da Sesame[49], una piattaforma open-source molto utilizzata per interrogare e analizzare i dati RDF. Il più potente linguaggio d'interrogazione che supporta è SeRQL, che è recentemente divenuto compatibile con SPARQL. Un'altra infrastruttura, di tipo peer-to-peer, che consente l'interrogazione di ontologie basate su RDF è Edutella[50], che utilizza un linguaggio d'interrogazione chiamato QEL. Avvicinandosi invece di più anche ai tools ragionanti sulle ontologie, troviamo Jena[51], un framework Java che oltre a supportare SPARQL come linguaggio d'interrogazione, supporta un ambiente sia per RDF che per OWL. Inoltre include alcune regole per ragionare su ontologie, basate su inferenze. Un altro potente strumento per interrogare e manipolare le ontologie RDF è TRIPLE[52]: è un linguaggio che consente di definire la semantica di vari linguaggi, attraverso la definizione di regole, e in particolare, le regole definite in TRIPLE permettono di ragionare su informazioni delle annotazioni RDF, dopo che esse sono state tradotte da RDF a TRIPLE. Infine altre due piattaforme per la gestione e il ragionamento su ontologie che si possono citare sono KAON2[53] e Flora-2[54]. KAON2 è una piattaforma per gestire ontologie OWL-DL, SWRL e ontologie basate su F-logic; Flora-2 è invece uno strumento object-oriented, il cui linguaggio è un dialetto di F-logic; le applicazioni di Flora-2 includono agenti intelligenti, Semantic Web, gestione ontologie, integrazione di informazioni e molte altre.

## 2.2 Tecnologie proposte per semantica debole e approcci ibridi

Per quanto riguarda la semantica debole, sono presentati due tipi di tecnologie: i Microformats[55] e RDFa[56]. Entrambe non si presentano come tecnologie molto potenti, e per certi versi ancora poco utilizzate, ma possono acquisire maggiore rilevanza se combinate o usate assieme a strumenti della semantica forte, e quindi nel campo di quelli che sono stati definiti approcci ibridi.

### Microformats

I Microformats sono una serie di formati per i dati, costruiti sugli standard esistenti del linguaggio di markup XHTML e HTML. Questi set di formati hanno il compito di introdurre una descrizione semantica dei dati nei documenti HTML e XHTML, e per fare ciò sono usati in combinazione con i loro

relativi attributi (*class* e *rel*). È una tecnologia ancora nuova e in fase di sviluppo, ma già le ultime versioni dei browser, grandi portali e motori di ricerca, come Yahoo! e Google, stanno iniziando ad implementarli. I principi base per cui sono stati costruiti tali formati sono:

- risolvere problemi specifici;
- essere il più semplice possibili;
- disegnati in primo luogo per l'uomo, e poi per le macchine;
- riutilizzare blocchi e costrutti già definiti dagli standard ampiamente adottati;
- modularità e incorporamento (*embeddability*);
- abilitare ed incoraggiare lo sviluppo decentralizzato di contenuti e servizi.

Alcuni esempi di Microformats al momento esistenti sono: *hCard*, che è un microformat per includere informazioni di *vCard* (un particolare formato usato per i biglietti da visita elettronici); *geo* è invece un formato per la rappresentazione delle coordinate geografiche (latitudine/longitudine), che possono essere utilizzate da strumenti come GPS, o mappe; *hCalendar* consente l'inclusione di informazioni su eventi, nel formato *iCalendar*; *hReview* è un microformat per includere revisioni e votazioni; *hAtom* utilizzato per il marking up dei feed Atom, utilizzato principalmente per i Webblogs; *XFN* similmente a FOAF è un formato utilizzato per rappresentare relazioni sociali; *rel-tag* infine è un formato per taggare porzioni dei contenuti: basta aggiungere *rel="tag"* ad un hyperlink che include il contenuto, e nell'ultima parte dell'URL viene specificato il tag per esso.

## RDFa

Un formato proposto invece per gli approcci ibridi è RDFa. Come i Microformats, anche RDFa si pone il compito di introdurre aspetti semantici nei documenti XHTML. RDFa è una recommendation del W3C, che più in particolare rispetto ai Microformats, permette di incorporare aspetti semantici descritti sotto forma di RDF nei documenti XHTML. In particolare per aggiungere tali estensioni semantiche, e per specificare l'URI di riferimento al modello dei dati, sono utilizzati gli attributi XHTML. Inoltre RDFa può essere utilizzato in combinazione con i microformats, per rappresentare determinate informazioni che non sono mappabili in essi (poiché magari non esiste un microformat per esse). Esistono inoltre alcuni standard del W3C

che permettono la trasformazione di dati da XML e XHTML in RDFa, e in modo particolare per convertire da microformat a RDFa.

## Capitolo 3

# Funzionamento dei SASs

Si analizza ora come funzionano i SASs. A questo scopo è utile definire e descrivere in primo luogo un modello funzionale per essi. Tale modello è costituito principalmente da tre componenti, che sono rispettivamente:

- **Acquisizione:** è la componente con cui il sistema raccoglie informazioni riguardanti l'utente e il contesto;
- **Gestione della conoscenza:** è la componente che gestisce ed elabora i dati e le informazioni acquisite. Queste informazioni sono rappresentate e memorizzate all'interno dei modelli di dominio, contesto e utente. Siccome il modello funzionale dei SASs è iterativo, questa componente si occupa di aggiornare, estendere e modificare tali modelli, in relazione alle nuove informazioni acquisite. Quest'estensione della conoscenza può essere attuata anche mediante tecniche di inferenza e ragionamento;
- **Produzione dei processi di adattamento:** si attua l'adattamento del sistema in relazione ai modelli e alle informazioni in essi contenute.

Questo modello è iterativo, ovvero le componenti sopra descritte operano in modo ciclico, ripetendosi più volte nel tempo, e ad ogni iterazione si aggiorna l'adattamento, come mostrato in figura 3.1. Seguendo le tre componenti di questo modello si andranno ora a descrivere più in dettaglio i suoi vari aspetti.

### 3.1 Acquisizione e scambio dei dati tra SASs

Come prima cosa ci si chiede come i SASs acquisiscono dati ed informazioni che saranno in seguito utilizzate per la modellazione e che guideranno il processo adattivo. In particolare si andrà ad analizzare come sono generalmente

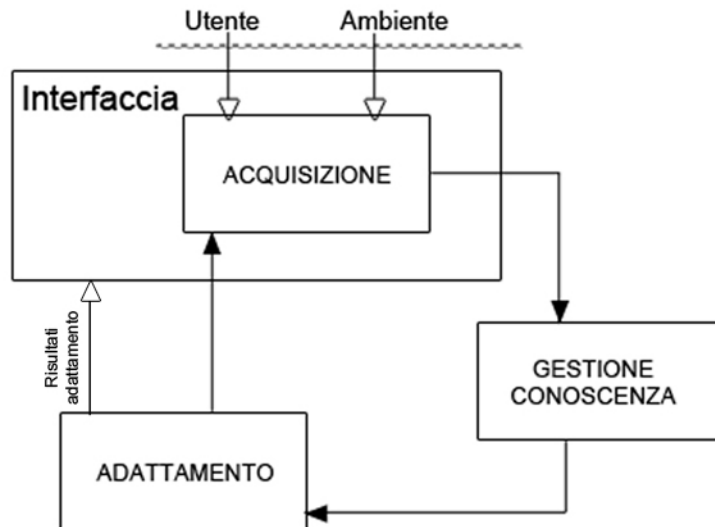


Figura 3.1: Schema descrittivo del funzionamento dei SASs.

rappresentate tali informazioni, in quale modo sono acquisite ed eventualmente scambiate con altri sistemi (integrazione). Infine si vedranno i formati utilizzati per rappresentare i dati.

### 3.1.1 Rappresentazioni utilizzate per i tag

I dati e le informazioni utilizzate dai SASs, sono rappresentati per mezzo di “tag” o annotazioni. Tali tag sono esprimibili e descrivibili mediante molti formati messi a disposizione dalle tecnologie del Semantic Web e di Web 2.0.

Ad esempio essi potrebbero essere rappresentati con statement RDF, ovvero per mezzo di triple “risorsa-proprietà-valore” (usualmente con sintassi XML). Mediante l’utilizzo di istanze di alcuni RDF Schema, come ad esempio:

- informazioni riguardanti le caratteristiche di un utente possono essere descritte con l’istanza di un modello FOAF (figura 3.2). I tag in questo caso rappresentano i valori associati alle proprietà del modello FOAF, come per esempio alla proprietà “name” è associato il nome dell’utente, o alla proprietà “mbox” la sua e-mail.
- mediante l’istanza di un modello Dublin Core, si possono descrivere informazioni riguardanti una qualche risorsa. Analogamente a quanto detto per FOAF, i tag questa volta rappresentano i valori associati agli

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:Person>
    <foaf:name>Jhon</foaf:name>
    <foaf:surname>Doe</foaf:surname>
    <foaf:mbox rdf:resource="JhonDoe@JhonDoe.com" />
  </foaf:Person>
</rdf:RDF>

```

Figura 3.2: Esempio di descrizione FOAF

elementi del modello Dublin Core. Per esempio all'elemento "creator" verrà attribuito il nome del creatore della risorsa, all'elemento "title" verrà associato il titolo o nome attribuito alla risorsa.

Questi sono solo due esempi, ma moltissimi altri schemi sono oggi disponibili per descrivere informazioni provenienti da svariati domini.

Altri formati utilizzabili per rappresentare i tag sono Microformats o RDFa, per descrivere ed integrare informazioni nell'HTML o XHTML. Un semplice esempio di utilizzo di Microformats potrebbe essere l'uso di hCard da attribuire alle informazioni sui contatti di una pagina Web (figura 3.3).

```

Per esempio, si consideri l'informazione sui contatti

<div>
  <div>Joe Doe</div>
  <div>The Example Company</div>
  <div>604-555-1234</div>
  <a href="http://example.com/">http://example.com</a>
</div>

Con il microformat hCard, diventa:

<div class="vcard">
  <div class="fn">Joe Doe</div>
  <div class="org">The Example Company</div>
  <div class="tel">604-555-1234</div>
  <a class="url" href="http://example.com/">http://example.com</a>
</div>

```

Figura 3.3: Esempio di uso del Microformat hCard

In modo simile può essere utilizzato anche RDFa, con la differenza che all'interno del documento XHTML vengono integrate proprietà di RDF o RDFS (dunque si possono integrare anche proprietà ed elementi di Dublin Core, di FOAF, e altri schemi).

Le informazioni utilizzate dai SASs, rappresentate per mezzo di tag, possono essere acquisite in vari modi dal sistema:

- i tag possono essere creati e posti dall'autore di una determinata risorsa utilizzata dal sistema (questo nell'ambito del Semantic Web: l'autore marca una sua risorsa con una descrizione che appare in un'ontologia);
- i tag possono essere creati e posti da utenti, su risorse utilizzate dal sistema (e quindi un approccio di Web 2.0, in cui sono i vari utenti e utilizzatori del sistema che marcano le risorse; utilizzo dunque di folksonomie);
- i tag possono essere acquisiti da altri sistemi (e quindi mediante uno scambio di informazioni).

Alcuni sistemi che utilizzano i primi due tipi di approcci per acquisire informazioni sono:

- iFanzzy[9] (classificato come SAS forte) è un sistema che offre una personalizzazione dei programmi tv per i suoi utenti, ed è basato su un ontologia, chiamata “MPEG-7 ontology”, che definisce varie proprietà e informazioni per modellare programmi e canali televisivi .
- Group Me![58] è un sistema (classificato come SAS debole) in cui agli utenti è permesso di caricare oggetti multimediali (dunque foto, video, testi, ecc.) e di “taggarli”; i tag sono in seguito trasformati in formato RDF, e possono essere esportati ed utilizzati da altri sistemi.

Si approfondiscono nel prossimo paragrafo le tecniche che riguardano lo scambio dei tag.

### 3.1.2 Tecniche per gestire lo scambio di tag tra SASs

I dati e le informazioni con cui lavorano alcuni SASs possono provenire ed essere scambiati con altri sistemi, generando così una situazione di interoperatività tra sistemi adattivi. Usualmente vengono scambiati dati integrati, costituiti da informazioni provenienti da domini e tipi di conoscenza differenti. Tale scambio è attuabile in modo semplice e agile, grazie alla grande flessibilità di RDF, che permette di descrivere e rappresentare diversi tipi di conoscenza con lo stesso formato. Sempre grazie a questo vantaggio portato da RDF è possibile effettuare tali scambi di informazioni tra i vari sistemi in modo automatizzato. Tuttavia si possono presentare alcune problematiche legate allo scambio dei dati, in quanto ogni sistema definisce le proprie ontologie. Affinché i dati scambiati siano comprensibili e utilizzabili dagli altri



sistemi, si rende necessario per ognuno di essi gestire l'eterogeneità semantica dei tag e delle annotazioni scambiate, che si riferiscono a concetti espressi in ontologie diverse.

Le soluzioni proposte a tale problema sono quelle di allineare, fare un matching, o fondere assieme le ontologie. L'allineamento (della conoscenza) può essere automatico, semi-automatico o manuale. Esempi significativi di SASs che utilizzino un allineamento automatico del dominio di ontologie non sono ancora stati sviluppati, per quelle che sono le attuali conoscenze dell'autore, tuttavia sono stati proposti alcuni algoritmi per attuare tale processo di allineamento. Sono disponibili invece esempi di SASs che attuano un allineamento manuale o semi-automatico delle ontologie, come HERA[59] e iFanzzy. Il primo definisce un modulo di integrazione del modello delle ontologie, utilizzato dai progettisti per creare i collegamenti tra i concetti presenti in esse. Il secondo invece trasforma le informazioni in formato OWL/RDF e utilizza SKOS per descrivere relazioni tra vari concetti. Un'altra soluzione può essere attuata grazie l'utilizzo delle "reference ontologies", ossia l'ontologia propria di un dato sistema viene "mappata" in una specifica ontologia di riferimento, che può essere ritenuta standardizzata, e quindi compresa e riconosciuta da altri sistemi. Un esempio utile per comprendere questo concetto può essere quello di PRF(Personal Reader Framework)[10], un sistema che fornisce un ambiente per la progettazione e l'esecuzione di servizi personalizzati, in grado di comprendere le annotazioni RDF relative a varie risorse. La comprensione di tali annotazioni, che possono provenire da file RDF importati nel sistema dall'esterno, avviene mediante una sorta di "binding RDF", in cui l'ontologia che li descrive viene mappata in ontologie considerate standard (relativamente al campo dell'e-Learning) come LOM e IMS, usate per descrivere metadati e risorse. Una volta mappata l'ontologia il sistema adattivo la può utilizzare con efficacia.

Un'altra caratteristica importante riguardante l'interoperabilità tra SASs è data dalla gestione dello scambio di risorse annotate. Tale gestione generalmente è attuata utilizzando la tecnologia dei servizi Web. Per molti SASs al fine di definire una semantica per la loro interoperatività, è importante avere una descrizione semantica dei servizi Web che possono offrire. A tale scopo sono molto utili quelle tecnologie precedentemente illustrate, riguardanti i linguaggi e le ontologie che descrivono i servizi del Semantic Web, come WSDL, OWL-S e altri. Grazie a tali strumenti alcuni SASs possono personalizzare i propri servizi, e attuare dunque anche un adattamento delle proprie funzionalità.

Il sistema PRF è un ottimo esempio di questo approccio: infatti, PRF utilizza WSDL per descrivere vari servizi, ognuno dei quali mette a disposizione

delle funzionalità adattive, come ad esempio raccomandare oggetti d'apprendimento (quindi risorse o testi da leggere, o studiare), suggerire esercizi, ecc. ognuno di questi servizi è personalizzato a seconda dell'utente che lo utilizza. Dunque PRF mette a disposizione una sorta di framework per un e-Learning personalizzato.

Il concetto di servizio Web personalizzato offerto da PRF, può essere però esteso anche al di là dell'e-Learning, a domini più generici e differenti. Ad esempio il sistema MyEar[60], sfrutta il servizio Web dei feed RSS<sup>1</sup>. MyEar genera dei podcast<sup>2</sup> personalizzati secondo i gusti musicali di ogni utente, e tramite il servizio dei feed RSS li mette a disposizione dei suoi utenti. MyEar è a sua volta basato su MyNews[60], un altro servizio Web analogo, basato però sulla creazione di feeds personalizzati a proposito di notizie.

Un altro esempio di sistema che utilizza questo genere di strumenti per la gestione dei servizi Web è COMPASS[13], un'applicazione mobile per turisti. COMPASS è una piattaforma aperta, che permette ad altri sistemi di integrare in essa nuovi servizi, pubblicandoli in un apposito registro dei servizi Web (descrivendoli quindi tramite WSDL), e di correlarli con la rispettiva ontologia (estendendo quindi il loro significato semantico, con un ontologia OWL), in un registro delle ontologie. Per esempio: un'organizzazione possiede un servizio Web che mostra agli utenti una collezione di vecchie cartoline digitalizzate. Quest'organizzazione se vuole può integrare questo servizio su COMPASS, pubblicandolo nell'apposito registro dei servizi Web, e pubblicando anche la relativa ontologia che lo descrive, nel registro delle ontologie.

Infine un ultimo esempio di questa tipologia di sistemi che si può citare è SmartWeb[61]. Si tratta di un progetto mirato allo sviluppo di interfacce multiuso per comporre e distribuire Semantic Web Services su piattaforme mobili. I servizi Web sono descritti mediante OWL-S, e le ontologie sono utilizzate per rappresentare il significato semantico delle query degli utenti e per comporre le risposte da parte dei vari servizi Web.

Quanto detto finora, riguarda principalmente i SASs forti, in quanto si è parlato di ontologie e di Semantic Web. Lo scambio di tag e informazioni per SASs deboli è in principio più semplice, perché soggetto a meno vincoli, ma può essere riconducibile allo scambio di informazioni tra SASs forti, più precisamente grazie all'uso di approcci ibridi. Infatti, molti dei SASs deboli,

---

<sup>1</sup>feed RSS: è un particolare formato per raccogliere e distribuire sul Web notizie e informazioni (formato basato su XML). Il protocollo di scambio di informazioni in questo formato è utilizzato soprattutto da blog e sistemi che diffondono notizie e news, in cui le risorse diffuse sono aggiornate frequentemente.

<sup>2</sup>Podcast: sono documenti, generalmente in formato audio o video, diffusi utilizzando il protocollo di scambio dei feed RSS.

raccolgono le social annotation e i vari tag, e li trasformano in RDF. Questi dati possono poi essere scambiati con lo stesso meccanismo usato per i SASs forti. Un esempio di sistema che attua questa modalità è il già citato Group-Me!, il quale trasforma i tag posti sulle rispettive risorse multimediali in RDF, e permette la loro esportazione verso altri sistemi, che possono riutilizzarli. Un altro esempio viene da Revyu[62], che utilizza un approccio ibrido: fa uso di Microformats per esportare dati, mentre quelli che importa dall'esterno vengono trasformati in RDF tramite un RDF-binding e mantenuti in una repository; in tale modo i dati importati nel sistema possono essere internamente gestiti usando tecniche di semantica forte. Un altro meccanismo agile ed economico per scambiare tag e annotazioni è basato sull'uso di feed RSS, di cui un esempio sarà fornito in seguito (vedi paragrafo 3.2.2, pagina 32).

## 3.2 Gestione della conoscenza

Il passo successivo all'acquisizione di dati e informazioni è la loro gestione ed elaborazione. E' necessario per i SASs creare dei modelli che descrivono come immagazzinare i dati e le informazioni acquisite. Tali modelli possono in seguito essere estesi o modificati con l'acquisizione di nuove informazioni. Questa è una fase molto importante poiché molti sistemi si basano proprio sull'estensione della conoscenza per svolgere il processo adattivo, ovvero prendono in considerazione questi modelli e le loro estensioni, le nuove informazioni acquisite, e a partire da esse generano l'adattamento. I SASs che utilizzano questo modello di funzionamento possono fare riferimento a vari tipi di conoscenza:

- conoscenza sul dominio: esprime concetti del dominio, i suoi contenuti e servizi;
- conoscenza sull'utente: esprime le caratteristiche di un utente, la sua conoscenza, i suoi interessi e preferenze;
- conoscenza del contesto e dei dispositivi: definisce il contesto in cui l'applicazione lavora;

Si possono poi specificare tipi di conoscenza più dettagliata, riguardo all'utente e al contesto. Relativamente all'utente si ha:

- conoscenza delle interazioni dell'utente col sistema;
- conoscenza degli obiettivi e dei piani dell'utente;
- conoscenza delle relazioni sociali date dai social network, sull'utente;

- conoscenza sul modo di descrivere le attività: descrive come l'utente esegue varie attività.

Relativamente al contesto e ai dispositivi:

- conoscenza sul modello di presentazione e navigazione degli hypermedia;
- conoscenza sul modello d'adattamento: specifica le regole che descrivono il comportamento adattivo del sistema;
- conoscenza del modello d'integrazione: informazioni su come interagiscono i vari sistemi per scambiarsi informazioni.

Al fine di comprendere come vengono gestiti ed elaborati dai SASs questi diversi tipi di conoscenza, è necessario conoscere e avere un'idea di come sono fatti i modelli che la contengono e la descrivono. Per questo saranno mostrati alcuni aspetti della modellazione di dominio, contesto e utente, che sono i tre modelli utilizzati dai SASs, ricordando però che la modellazione non fa parte del modello di funzionamento dei SASs, ma è una fase del loro sviluppo. Dunque si analizzano ora nel dettaglio gli aspetti di modellazione del dominio, del profilo utente e del contesto, e della gestione di tali modelli. In particolare si mostrerà come questi modelli possono essere estesi sia grazie a nuove informazioni scambiate con altri sistemi, che mediante tecniche di ragionamento e inferenza sui modelli stessi.

### **3.2.1 Modellazione del dominio, del contesto e dell'utente**

#### **Modellazione del dominio**

La modellazione del dominio permette di descrivere un determinato tipo di conoscenza, con i relativi concetti e contenuti specifici. In particolare tale modellazione è utile per rappresentare i concetti che saranno usati per annotare e indicizzare risorse. Nella parte riguardante l'acquisizione delle informazioni, le risorse sono annotate e indicizzate o attraverso concetti che fanno riferimento a ontologie (per i SASs forti), o attraverso tag posti su esse dagli utenti finali (per i SASs deboli). Il compito del modello è descrivere la struttura di questi dati. Tale modellazione, almeno relativamente ai SASs forti, può essere gestita utilizzando RDF e RDF Schema. Per codificare questi modelli RDF e RDFS si utilizza in genere la sintassi XML, che grazie alla sua grande flessibilità ed estendibilità consente di definire abbastanza liberamente i tag e marcatori per descrivere vari elementi di domini differenti. Il formato

RDFS consente inoltre di descrivere la semantica degli elementi del dominio, definendo le proprietà e le relazioni semantiche che li collegano, costruendo così un'ontologia, o magari espandendo i concetti di una già in utilizzo. L'uso di XML per modellare la semantica delle componenti di un dominio è stato scartato, poiché è risultato poco flessibile per rappresentare questo tipo di conoscenza. Questo è dovuto al fatto che XML è un linguaggio di modellazione, adatto a modellare dati e documenti, mentre RDF è un linguaggio di modellazione di metadati, dunque più adatto per modellare la conoscenza (le relazioni semantiche) relativa alle varie risorse e documenti.

### **Modellazione dell'utente**

Al fine di raccogliere e poter gestire informazioni e caratteristiche relative ai vari utenti è necessario un modello del profilo dell'utente. Tale modello, analogamente a quello del dominio, può essere gestito sempre con RDF e RDF Schema. In particolare FOAF è uno schema RDF molto utilizzato per questo scopo. In un modello FOAF infatti, oltre che alle informazioni comuni di un utente (come nome, cognome, numero di telefono, e-mail, ecc.) è possibile memorizzare al suo interno informazioni relative agli interessi dell'utente, e alle sue relazioni sociali con altri utenti (come ad esempio gli altri utenti che conosce). Un altro modello, disponibile invece per i SASs deboli, è l'uso di Microformats come ad esempio hCard, utile per rappresentare e modellare informazioni relative a persone, compagnie ed organizzazioni. Infine per i SASs ibridi si può utilizzare RDFa (presentato nel paragrafo 2.2, pagina 18), con il quale per esempio è possibile inglobare nelle pagine XHTML dei vari utenti un loro modello FOAF.

### **Modellazione del contesto**

Il modello del contesto esprime informazioni riguardanti l'ambito e l'ambiente in cui il sistema funziona, e dunque gestisce informazioni provenienti dall'esterno come ad esempio il tipo di device utilizzato per interfacciarsi col sistema, o il tipo di utente che lo sta utilizzando, oppure informazioni riguardanti le componenti esterne che gli inviano determinati dati, ecc. Il modello del contesto risulta fortemente legato col modello dell'utente, tanto che a volte entrambi i modelli sono integrati assieme e considerati come uno unico. In alternativa, questo legame tra i modelli può far sì che determinate caratteristiche dell'utente siano considerate nel modello del contesto, o viceversa, caratteristiche del contesto sono considerate nel modello dell'utente.

### **3.2.2 Estensione della conoscenza dei modelli di dominio, contesto e utente**

Come illustrato in precedenza, i SASs possono estendere la loro conoscenza relativa al dominio, al contesto e all'utente per mezzo di due tecniche: attraverso l'integrazione di dati e informazioni scambiate con altri sistemi, oppure per mezzo di tecniche di ragionamento ed inferenza sui dati già in loro possesso.

#### **Estensione del modello del dominio con l'integrazione dei dati con altri SASs**

Una delle caratteristiche più importanti, legata al modello del dominio, è la possibilità di estendere i suoi contenuti, per mezzo di risorse e nuove informazioni provenienti dall'esterno o da altri sistemi. Queste ulteriori informazioni vanno ad arricchire il modello del dominio, aumentando la conoscenza del sistema. Tra i sistemi che utilizzano queste tecniche di espansione della conoscenza, è bene citare gli "Open Adaptive Hypermedia Systems" (OAHS). Questi sistemi, grazie all'utilizzo di risorse esterne, generano una personalizzazione dei servizi o dei contenuti anche in scenari aperti e sono in grado di fornire tecniche di adattamento anche al di fuori del contesto locale: AHA!<sup>[3]</sup> rappresenta un'infrastruttura per costruire questi tipi di sistemi. Più avanti, saranno descritti più in dettaglio.

Per quanto riguarda i SASs deboli, talvolta l'uso di tecniche semantiche deboli per la gestione e l'espansione del dominio può risultare preferibile e più vantaggioso. Infatti, grazie a esse è possibile gestire domini instabili, soggetti a rapidi cambiamenti, in cui le risorse sono annotate e indicizzate da tag posti dai singoli utenti. In questo modo i tag risultano soggettivi e relativi a categorie magari nuove, non ancora ben definite o in evoluzione. Tuttavia è bene ricordare che le tecniche semantiche deboli sono meno potenti, e quindi hanno un potere semantico inferiore rispetto a quelle forti. Per questa ragione risultano più validi approcci ibridi.

#### **Estensione del modello dell'utente con l'integrazione dei dati con altri SASs**

Il modello dell'utente può essere creato e gestito in modi diversi e presentarsi a volte molto differente da un sistema all'altro. Per questo motivo, una delle problematiche principali, che si presenta sia nei SASs forti che in quelli

deboli, riguarda lo scambio e l'integrazione di modelli del profilo utente. Oltre a questo ci sono problematiche legate alla privacy e alla sicurezza dei dati contenuti nel sistema adattivo.

Partendo dai SASs forti, la prima problematica implica il dover risolvere eventuali conflitti e contraddizioni tra valori relativi a un certo utente, se i dati sono acquisiti da sistemi diversi. Tali conflitti potrebbero sorgere a causa di una manipolazione imprecisa dei valori, oppure perché alcuni sistemi possono contenere dati vecchi relativi ad un utente, che quindi non sono aggiornati. Altre cause della nascita di tali conflitti possono essere che sistemi diversi attuano in genere inferenze diverse sui dati, o per altri motivi meno frequenti. Tale problema si può suddividere a sua volta in altre tre problematiche più specifiche:

- problema della rappresentazione delle caratteristiche dell'utente;
- problema dell'identificazione dell'utente;
- problema della gestione dello scambio di profili utente.

Il problema della rappresentazione delle caratteristiche dell'utente dipende dal fatto che caratteristiche diverse degli utenti sono modellate da sistemi differenti, ognuno dei quali utilizza le proprie ontologie.

Una prima soluzione proposta a questo problema viene da Heckmann[22], che propone l'uso di una reference ontology da usare come standard, per vari sistemi, per mappare le proprie ontologie riguardanti la modellazione degli utenti. In particolare, tale ontologia definita da Heckmann è chiamata GUMO (General User Model Ontology), e rappresenta dunque, il modello utente in un ambiente generico, in modo standard. Tale ontologia è basata su "situational statements", ovvero statements che contengono informazioni riguardanti le caratteristiche utente del modello, dati relativi al contesto o dati rilevati durante l'interazione dell'utente col sistema. Nella soluzione di Heckmann questi statements sono scambiati grazie ad UserML, uno specifico linguaggio che estende le triple RDF con settuple, mettendo a disposizione quattro slot in più in cui si possono memorizzare informazioni riguardanti restrizioni spaziali e temporali sui dati scambiati, metadati riguardanti il proprietario delle informazioni, informazioni sulla privacy e confidenzialità. L'idea è quella di aggiungere informazioni ai dati scambiati di ogni utente, per limitare i conflitti, aumentare il potere espressivo degli statements in relazione alle situazioni in cui sono scambiati, favorendo così la loro integrazione in altri sistemi. Analizzando brevemente i pro e i contro di questo approccio, si può dire che ha il vantaggio di risolvere gran parte delle problematiche riguardanti i conflitti e l'integrazione nel modello del profilo utente di dati

scambiati tra SASs differenti. Tuttavia presenta come difetto la necessità di imporre specifiche caratteristiche al dominio, per modellare propriamente il profilo utente. Un modello utente che memorizza specifiche caratteristiche del dominio, può presentare problemi quando viene scambiato tra i vari sistemi: le caratteristiche possono essere percepite e classificate in maniera diversa in domini differenti. Tale problema potrebbe essere semplicemente risolto con l'utilizzo di standard per modellare domini. Purtroppo degli standard veri e propri per questi modelli generici non esistono ancora. Esistono standard specifici, per determinati tipi di domini (come già citati IMS e LOM, per l'e-Learning), ma non standard per una modellazione "generica".

Un'altra soluzione proposta al problema della rappresentazione delle caratteristiche dell'utente è quella di lasciare ad ogni sistema la libertà di definire i propri modelli utente e le proprie ontologie, in base ai propri usi e necessità, ed in seguito utilizzare dei "mediatori" per risolvere eventuali conflitti tra i modelli, e gestire l'eterogeneità dei dati. Tale approccio, proposto da Berkovsky et al.[17] è chiamato "mediazione dei modelli utente". L'idea è che i vari conflitti che si possono presentare, sono risolti da un mediatore, che fa uso di tecniche di ragionamento e meccanismi d'inferenza. Un esempio di mediatore è dato da GUC (Generic User model Component): questo sistema genera dei mappings tra gli schemi dei modelli utente coinvolti nello scambio tra SASs. Questi mappings sono creati manualmente dai progettisti ed espressi come regole SWRL, che hanno il compito di gestire eventuali conflitti tra i modelli scambiati. In seguito, ogni volta che avverrà uno scambio di modelli utente tra i SASs coinvolti, saranno applicate tali regole. Un altro esempio di mediatore si può trovare nel lavoro di Carmagnola e Dimitrova[14], in cui lo scambio dei dati relativi agli utenti è gestito automaticamente da un algoritmo che computa le similarità semantiche tra classi di utenti e ontologie differenti.

Il secondo problema menzionato è quello relativo all'identificazione dell'utente. Principalmente tale problema si presenta e risulta rilevante solo per SASs che operano in scenari vasti e aperti, nei quali l'identificazione dell'utente viene fatta attraverso le sue interazioni col sistema stesso, e attraverso informazioni relative all'utente memorizzate in esso. Una soluzione a tale problema è data dall'appena menzionato framework di Carmagnola e Dimitrova: in tale approccio l'utente è considerato come una collezione di proprietà e caratteristiche, e l'algoritmo di Carmagnola e Dimitrova consente di riconoscere i vari utenti attraverso sistemi differenti analizzando i valori attribuiti a caratteristiche comuni tra i vari profili. Un altro approccio a tale problema può essere ancora l'utilizzo di reference ontology. Un esempio di utilizzo di tale ontologia è dato da ELENA[21]. Questo sistema identifica un utente in base



a varie annotazioni riferite a un'ontologia che descrive gli studenti in modo standard; oppure un'altra ontologia, che può essere usata come reference ontology, utile per l'identificazione degli utenti è FOAF.

Infine la terza problematica, relativa alla gestione dello scambio di profili utenti, viene trattata seguendo due approcci principali: l'utilizzo di Semantic Web Services, che mediante API pubbliche esportano i valori e caratteristiche dell'utente; oppure l'utilizzo di infrastrutture semantiche per interrogare le caratteristiche e i valori riguardanti i vari utenti. Un esempio di sistema che utilizza il primo approccio è dato dal framework CHIME[3], che estende il sistema AHA!. Ogni componente di tale sistema è vista come un servizio, ed ogni servizio è definito sulla base di un'ontologia specifica. In particolare il "Broker-based Discovery Service" è uno dei servizi Web definiti da CHIME, che consente ad AHA! di scambiare i modelli utente con altri sistemi. Come esempio invece del secondo tipo di approccio, si può citare Edutella, un'infrastruttura peer-to-peer che consente di interrogare modelli RDF per estrarre dati e informazioni, e quindi in questo caso particolare, caratteristiche e valori dai modelli degli utenti.

Per quanto riguarda i SASs deboli invece, sono pochi quelli che effettivamente scambiano e integrano tra loro informazioni sia sul contesto che sull'utente, e generalmente fanno uso di approcci misti. Si scambiano i tag generati dagli utenti e li mappano in una qualche ontologia di loro utilizzo, oppure li utilizzano per descrivere una data risorsa, o da usare come "prova" dell'interesse dell'utente verso quella risorsa, accrescendo così il modello utente con informazioni riguardanti i suoi interessi.

Il problema dell'identificazione automatica di un utente effettuata da SASs differenti, è stato affrontato solo da un SAS: Google Social Graph API[63]. Analogamente al caso dei SASs forti, questo sistema prova a confrontare le caratteristiche di un utente messe a disposizione in sistemi differenti, per identificarlo univocamente. In particolare Google Social Graph API utilizza una tecnica di matching dei profili, basata sullo studio delle homepage degli utenti, se queste sono disponibili. Altri approcci semi-automatici sono stati proposti da SAS come CHIP[64] e iCITY[65], che richiedono agli utenti di sottoscrivere i feed RSS messi a disposizione dagli altri sistemi con cui interoperano.

CHIP e iCITY sono inoltre due validi esempi che trattano il problema della gestione dello scambio di tag e profili utente. Di fatto, questi due sistemi operano in modo simile: CHIP è un sistema che provvede a fornire una guida, ovvero, un tour personalizzato all'interno di un museo, che si adatta agli interessi dell'utente che lo sta utilizzando. iCITY similmente è un sistema

che organizza una guida personalizzata, in relazione agli interessi dell'utente, ma attraverso eventi culturali di una città (quella di Torino). Entrambi questi sistemi possono importare ed esportare feed RSS contenenti tag che si riferiscono agli interessi degli utenti. Sarebbe dunque possibile realizzare l'interoperatività tra essi mediante lo scambio di feeds. Tuttavia anche in questo esempio, come in uno degli approcci visti per i SASs forti, risulta necessario un "mediatore", che gestisca l'eterogeneità dei tag e delle annotazioni. Un valido mediatore proposto è il già citato SKOS. Da questo esempio emerge che i feed RSS possono essere un valido strumento per gestire lo scambio di tag e annotazioni per i SASs deboli.

Infine un'altra soluzione, più vicina agli approcci misti, è quella di utilizzare una reference ontology. Ad esempio, si può utilizzare GUMO per attribuire un significato preciso ai tag scambiati che riguardano l'utente e il contesto. Mappando i tag si cerca di attribuire ad essi un significato semantico descritto in una qualche ontologia utilizzata dal sistema.

Per le problematiche concernenti la privacy sono state proposte alcune soluzioni, ma attualmente nessun SAS ha mai implementato qualcosa, o gestito in modo rilevante tale problema.

### **Estensione del modello del contesto con l'integrazione dei dati con altri SASs**

Le problematiche appena illustrate per il modello del profilo utente, si presentano in modo analogo per quello del contesto. Similmente ai modelli di dominio e utente, la problematica principale risiede nello scambio e nell'integrazione dei dati relativi al contesto tra i vari sistemi. Una soluzione a questo problema è offerta sempre da Heckmann[22], che definisce il modello UbisWorld[66]. Quest'ultimo utilizza GUMO e altre sei reference ontology per modellare informazioni sul contesto e sull'utente, e che possono essere scambiate sotto forma di situational statements. Un esempio di sistema che utilizza tale modello (UbisWorld) è PEACH, una sorta di guida per musei, simile a CHIP.

Un altro problema, legato alla gestione del modello del contesto, riguarda la necessità di una rapida acquisizione delle informazioni che si riferiscono al contesto stesso. Tale problema è affrontato dal progetto MobiLife[16], in cui varie informazioni sul contesto sono condivise e scambiate tra diversi sistemi, e integrate tra loro al fine di avere una conoscenza più precisa e generale sul contesto di utilizzo.

Infine un ultimo approccio proposto per la gestione dell'interoperatività è l'introduzione delle "Context Passport". Come dice il termine stesso si tratta di una sorta di "passaporti", che accompagnano gli utenti nella navigazione attraverso vari sistemi, venendo scambiati al passaggio da un sistema ad un altro. In pratica viene utilizzato un unico modello sia per i dati del contesto sia per quelli dell'utente (Unified User Context Model – UUCM), che viene scambiato tra i vari sistemi. In questo modo quando l'utente si va ad interfacciare ad un nuovo sistema, quest'ultimo acquisisce il modello UUCM, estrae le informazioni che gli occorrono su utente e contesto, e le utilizza per supportare le sue attività. Esempi di modelli UUCM sono UbisWorld o il modello utilizzato da GUMO, che integrano assieme informazioni sia riguardanti l'utente che il contesto.

### **Estensione di tali modelli per mezzo di ragionamenti**

Il secondo modo per estendere la conoscenza a riguardo di dominio, contesto e utente è l'utilizzo di tecniche di ragionamento. Più precisamente per i SASs forti saranno ragionamenti fatti sulla conoscenza offerta dalle ontologie. Per i SASs deboli invece si baseranno sulle folksonomie e sui tag posti dai vari utenti.

Partendo dai primi, si andranno ad analizzare queste tecniche di ragionamento sulle ontologie per inferire nuove informazioni. Il metodo base con cui costruire ragionamenti sulle ontologie è di fare deduzioni basilari per rendere esplicite alcune conoscenze ontologiche implicite. Tuttavia questo metodo spesso comporta ragionamenti difficili e controversi, e a volte non è sufficiente per acquisire nuova conoscenza. Si rende dunque necessario l'utilizzo dei linguaggi per definire regole semantiche (come RuleML, SWRL, e altri), per creare appunto le regole con cui esprimere i ragionamenti ontologici. Se è necessaria l'interoperabilità tra sistemi, tali regole possono poi essere scambiate per mezzo di altri linguaggi come R2ML e RIF. Se invece l'interoperabilità non è richiesta, si possono utilizzare in combinazione con linguaggi per definire regole non semantiche. Un esempio di sistema che utilizza un reasoner di questo tipo è ec(h)o[67], il quale carica le proprie ontologie DAML+OIL in Jess, una rule engine di Java, che permette di esprimere regole nel proprio linguaggio (e quindi Jess) oppure in XML. In particolare ec(h)o non tiene in considerazione regole semantiche espresse con RuleML fino a che non gli è richiesta l'interoperatività con altri SASs. Altri esempi di applicazione di ragionamenti sulle ontologie si trovano nel campo dell'e-Learning, in cui spesso tali ragionamenti sono utilizzati per comprendere e stabilire il livello di conoscenza dei concetti da parte degli utenti. Un esempio di tali siste-

mi è OntoAIMS[18], un sistema per l'e-Learning che fornisce informazioni sulle ricerche e sull'ambiente di apprendimento, sui compiti e gli obiettivi da raggiungere dagli studenti e sui vari concetti e risorse da far apprendere. La modellazione dell'utente in OntoAIMS è basata sul significato delle ontologie e su un dialogo interattivo con l'utente. Più precisamente, il sistema acquisisce informazioni interagendo con l'utente, attribuisce a esse un significato semantico in relazione alle opportune ontologie che utilizza, e da esse prova ad inferire nuove informazioni e conoscenze per mezzo di regole di ragionamento. Dopodiché nel dialogo con l'utente verifica se quanto dedotto è corretto oppure no, chiedendolo direttamente all'utente, ed in base al feedback che gli ritorna riesce a comprendere la correttezza e a verificare i suoi ragionamenti. Tali ragionamenti sono utilizzati inoltre per portare avanti il dialogo con l'utente; in altre parole le informazioni ricavate da essi, se corrette, sono poi riutilizzate per inferirne di nuove. Questa caratteristica di OntoAIMS mette in luce anche un'altra caratteristica dei ragionamenti, ovvero la possibilità di inferire nuova conoscenza su di un concetto a partire dalla conoscenza inferita a sua volta su altri concetti, con la possibilità dunque di creare una sorta di "catena" di ragionamento. Infine l'uso di tecniche di ragionamento può risultare utile per risolvere eventuali conflitti che si possono rilevare tra informazioni acquisite. Un esempio di quest'utilizzo si può trovare in PERSONAF[19], un framework che applica ragionamenti su ontologie riguardanti "case intelligenti e le persone al loro interno". Il modulo di ragionamento di tale sistema (ONCOR) è costruito come un set di risolutori, per interpretare tutti i tipi di conoscenza messi a disposizione dalle ontologie di PERSONAF. In particolare, grazie ad alcuni di questi risolutori è in grado di risolvere ed affrontare problemi di conflitti tra prove rilevate: ad esempio può risolvere conflitti tra posizioni rilevate da due sensori che hanno identificato una persona o un oggetto in due locazioni diverse della casa.

Per quanto riguarda i SASs deboli invece, l'estensione per mezzo di ragionamento riguarda principalmente il modello dell'utente. Tali sistemi processano le informazioni e la conoscenza riguardante gli utenti per estendere i loro modelli e per creare servizi personalizzati. La personalizzazione deriva principalmente dagli interessi di ogni utente, che sono inferiti ed estrapolati ragionando sui tag generati e posti sulle risorse dall'utente stesso. Possono nascere tuttavia due problematiche principali utilizzando questa tecnica:

- gli utenti possono porre tag anche su risorse a cui non sono interessati;
- gli utenti possono porre tag su risorse usando tag personali, che talvolta non descrivono la risorsa (ad esempio possono esprimere un'opinione, un apprezzamento).

Il primo problema può essere considerato irrilevante, se si assume che un utente generalmente utilizza segnalibri per marcare le risorse che vuole recuperare e che gli interessano. Anche il secondo problema può essere irrilevante se si considerano tutti i tag posti su una risorsa, dai quali dopo un certo numero si estrae una tendenza prevalente che andrà a descrivere quella risorsa. Quest'ultimo problema risulta invece particolarmente rilevante se si considerano i singoli tag. Questo poiché per identificare l'interesse di un utente a partire dalle sue singole annotazioni, è necessario prima di tutto identificare la categoria a cui appartengono i tag che egli pone su di una risorsa, e dalla categoria si può comprendere poi l'eventuale interesse. Per approfondire questo aspetto saranno ora mostrati alcuni approcci per analizzare ed estrapolare informazioni dai tag, al fine di individuare e modellare in maniera opportuna gli interessi degli utenti. Vi sono due approcci principali:

- analisi cognitiva dei tag;
- tecniche di estrazione dallo spazio dei tag.

L'analisi cognitiva dei tag consiste nel mappare i tag degli utenti in un modello multi-dominio, come può essere per esempio Wikipedia. L'idea di funzionamento è abbastanza semplice: i tag principalmente sono usati per descrivere risorse d'interesse per l'utente, dunque ogni volta che un utente pone dei tag, questi vengono aggiunti ad una collezione (o "nuvola") dei tag ad esso associata, aggiornandola continuamente (se il profilo utente è condiviso da più SASs questi possono operare assieme per aggiornare la collezione coi tag rilevati da ognuno di questi sistemi). In seguito questi tag sono mappati con categorie definite in un modello multi-dominio, e dal matching di essi nel modello vengono estrapolate le categorie di interesse. I riferimenti (URIs) a queste categorie vengono poi salvati all'interno di un file FOAF (nell'esempio con Wikipedia nel file FOAF vengono salvati negli interessi i riferimenti agli URIs delle categorie di Wikipedia). Utilizzando tali categorie si possono poi fare varie inferenze sugli interessi dell'utente, esplicitando relazioni tra esse, come per esempio da una categoria di interesse trovarne altre più generiche, o più specifiche. In particolare l'ontologia di Wikipedia si può considerare un buon modello multi-dominio poiché è continuamente aggiornata, molto ampia (definisce molte categorie) e include acronimi e termini dello slang comune, che sono talvolta molto utilizzati nei tag generati dalle persone.

Le tecniche di estrazione sono invece tecniche che permettono di identificare ed estrapolare interessi riguardanti gli utenti a partire da vari tipi di informazioni correlate con i tag e l'azione di tagging. Più nel dettaglio si possono identificare le seguenti tecniche:

- estrazione di altre proprietà degli utenti;
- estrazione di informazioni da annotazioni sugli utenti;
- utilizzo di algoritmi basati sulla tassonomia dei tag;
- utilizzo di tecniche di estrazione collaborative.

L'estrazione di altre proprietà degli utenti è un approccio che permette di elaborare nuove caratteristiche e proprietà relative a un utente inferendole dai tag che egli pone. Più in particolare il “tagging” è visto da due punti di vista diversi: l'azione di porre tag fornisce informazioni a proposito degli interessi dell'utente nel contenuto annotato, l'inclinazione dell'utente a interagire col sistema, o la sua tendenza a organizzare i contenuti, e altri aspetti che si possono rilevare dalle sue azioni e modi di porre tag. L'altro punto di vista riguarda invece dal significato dei tag, utilizzato invece per inferire altre proprietà sull'utente, come per esempio la sua conoscenza dei concetti, il suo livello di fiducia nella comunità degli utenti, o la sua creatività. Tale approccio è in fase di sviluppo su iCITY, in cui queste caratteristiche appena descritte sono combinate col modello dell'utente.

L'estrazione di informazioni da annotazioni sugli utenti invece consiste nel fare inferenze a proposito degli interessi di un utente da tag e annotazioni che altri utenti esprimono su di lui. Quest'approccio è dunque attuabile solo in sistemi in cui possono essere espresse annotazioni da utente a utente, come ad esempio i social network, ed ha il vantaggio di poter attuare una modellazione passiva di un utente.

Un altro approccio proposto è basato su algoritmi che considerano tassonomie di tag. I sistemi che utilizzano tale approccio apprendono in modo automatico una tassonomia di tag, e in seguito sono in grado di determinare quanto fortemente un utente è associato con determinati tag della tassonomia. In questo modo, analizzando i gruppi tassonomici di tag associati a determinati concetti, si possono inferire altri concetti a cui l'utente è interessato, anche se questi non sono esplicitamente descritti nello spazio dei suoi tag.

L'ultimo approccio da considerare è l'utilizzo di tecniche collaborative, utili soprattutto per trovare similarità tra utenti. In tale approccio si considerano i giudizi (o i voti) che gli utenti esprimono su determinate risorse, e in base a essi sono creati set di utenti (insiemi di utenti che hanno dato giudizi comuni o simili). In questo modo emerge un giudizio comune, o comunque prevalente sulle risorse. Il punto debole di quest'approccio è rappresentato dal rischio che in questo modo non emergano i veri e propri interessi del singolo utente sulle risorse, ma semplicemente informazioni sulla sua similarità con altri utenti.

### 3.3 Processi di adattamento

La fase finale del modello di funzionamento dei SASs è data dal processo di adattamento. In questa fase il SAS ha già acquisito varie informazioni, le ha utilizzate per produrre o modificare i modelli di dominio, contesto e utente, e ha esteso tali modelli per mezzo di tecniche di ragionamento o integrando in essi altre informazioni provenienti dall'esterno, per esempio scambiate con altri SASs. Quello che gli rimane da fare è realizzare l'adattamento per mezzo di un processo adattivo. L'obiettivo globale dei processi adattivi è quello di produrre un effetto di adattamento che tenga in considerazione le caratteristiche dell'utente e del contesto, che rispetti eventuali criteri sulla privacy, e che abbia cura esso stesso della controllabilità del processo d'adattamento. Definito quest'obiettivo, saranno illustrati in seguito i vari tipi di processi adattivi.

#### 3.3.1 Adattamento dei contenuti e loro suggerimento

Il primo tipo di adattamento che si andrà ad analizzare è l'adattamento dei contenuti e la loro raccomandazione (o suggerimento), ovvero si andranno ad analizzare sistemi che adattano e adeguano i contenuti da esporre o proporre ai loro utenti, creando così una sorta di personalizzazione dei servizi. In particolare i SASs che attuano questo tipo di adattamento fanno uso di due tecniche semantiche:

- recupero di risorse: consiste nell'andare a recuperare determinate risorse o concetti, in particolare per i SASs saranno risorse d'interesse per l'utente, dalle quali estrarre contenuti;
- filtraggio di informazioni: rappresenta in pratica la naturale forma di adattamento dei contenuti, in cui gli elementi del dominio sono rappresentati semanticamente, e dai quali vengono filtrate determinate informazioni e concetti per produrre a sua volta nuovi contenuti personalizzati. Tale metodo è anche chiamato “content-based approach”.

Infine è stato proposto anche un nuovo approccio chiamato “social network-based approach”, che produce la personalizzazione dei contenuti sfruttando gli interessi e le relazioni sociali espresse nei social network.

Partendo dai SASs forti, il recupero di risorse è attuato tipicamente interrogando le ontologie del dominio, spesso utilizzando esplicite query relative all'utente (fatte mediante i linguaggi di interrogazione per ontologie come SPARQL o altri). Queste query possono essere espanse o ristrette a seconda

del profilo dell'utente e del contesto, e della strategia di recupero utilizzata. Per esempio se una query restituisce pochi risultati, il sistema può riformularne un'altra che richieda dei concetti ad un livello superiore, meno specifici, e dunque la query viene espansa; oppure inversamente, se si vogliono meno risultati, magari più specifici, la query viene ristretta. Una volta recuperati i concetti d'interesse, prima di passare all'estrazione dei contenuti o raccomandazioni, si possono applicare tecniche di ragionamento su essi, per estrarne ed inferirne dei nuovi. Tali tecniche di ragionamento sono analoghe a quelle illustrate in precedenza per i SASs forti, dunque si tratta di ragionamenti sulle ontologie. A questo punto dai concetti estratti ed elaborati applicando il filtraggio delle informazioni si possono produrre contenuti personalizzati o suggerimenti ad altri contenuti.

Due esempi di SASs forti che utilizzano quest'approccio, per personalizzare i contenuti e suggerirne nuovi, sono rispettivamente PRF (Personal Reader Framework) e CHIP. PRF è un framework che fornisce meccanismi per la creazione di servizi personalizzati, in particolare lavora nel campo dell'e-Learning. Tra i servizi di PRF si possono trovare molti esempi di adattamento dei contenuti, in particolare si può prendere come esempio il servizio "Personal Curriculum Planner". Tale servizio produce una sorta di curriculum, o piano di studio personalizzato per ogni utente. Il funzionamento del Personal Curriculum Planner è basato sull'interrogazione del modello dell'utente e del contesto per estrarne concetti relativi ai suoi obiettivi e al suo livello di conoscenza, che sono rappresentati in modo semantico con statements RDF (questo si identifica con la fase di recupero di risorse). Successivamente tali statements vengono elaborati da un reasoner che applica una serie di regole di ragionamento sul dominio di conoscenza delle ontologie. Infine i nuovi concetti prodotti sono utilizzati dal sistema per estrarre i contenuti (fase di filtraggio). Questi contenuti sono una serie di corsi e attività di studio consigliati all'utente, per aiutarlo a raggiungere gli obiettivi da lui prefissati. Anche il sistema già menzionato, CHIP (si ricorda è un sistema che fornisce una guida personalizzata all'interno di un museo), ora verrà analizzato nel suo lato adattivo. In particolare, in CHIP un utente deve esprimere una votazione su vari artefatti che visita, e ognuno di questi artefatti è descritto da varie proprietà che fanno riferimento ad una determinata ontologia. Quest'ultima descrive in modo semantico i vari artefatti del museo e le relazioni che vi sono tra loro. Ogni volta che l'utente vota un artefatto, il voto che egli fornisce è propagato a tutte le proprietà che descrivono l'artefatto in questione. In questo modo il sistema è in grado di guidare l'utente nella sua visita proponendogli artefatti con le stesse proprietà, o altre simili a quelle degli artefatti che ha votato in maniera positiva, con un voto alto (e quindi pre-



sumibilmente quelli che gli sono interessati o piaciuti di più). In particolare CHIP interroga le sue ontologie e applica tecniche di ragionamento su esse per andare a recuperare concetti riguardanti le relazioni tra le varie proprietà, che poi utilizza per trovare gli artefatti da suggerire.

Analizzando l'altro approccio proposto, il social network-based approach, si nota che è basato principalmente sullo studio del livello di fiducia tra utenti, per suggerire contenuti. In pratica analizzando i giudizi e i voti che un utente esprime verso altri utenti all'interno dei social network, il sistema elabora una serie di valori di fiducia che quell'utente ha verso gli altri. Tali valori di fiducia sono poi dati in pasto ad algoritmi di raccomandazione, che li sfruttano per suggerire e raccomandare contenuti verso gli utenti. TrustProject[68] è un esempio in cui viene utilizzata la TrustOntology, un'ontologia basata sul vocabolario FOAF, per calcolare un livello di fiducia personalizzato tra utenti rispetto a specifici argomenti. L'efficacia di tale ontologia è stata testata in modo concreto in FilmTrust, un SAS che personalizza i contenuti da mostrare ai suoi utenti, come possono essere informazioni riguardanti film o recensioni create da altri utenti. In FilmTrust ogni utente può scrivere recensioni a proposito di un film, votare altre recensioni ed esprimere commenti su esse. Da tali azioni il sistema acquisisce informazioni sugli interessi dell'utente e sulla sua fiducia verso gli altri utenti, e in base ad esse personalizza i contenuti. Per esempio: se un utente A legge una recensione su di un film fatta da un utente B che non gli piace, esprimerà su di essa un giudizio negativo (o gli darà un voto basso). Questo andrà ad influenzare negativamente il livello di fiducia che l'utente A avrà verso l'utente B, quindi magari il sistema eviterà di proporre ad A altre recensioni fatte da B. Inversamente se A apprezza una recensione, o dei commenti che B fa su determinati film, questo accrescerà i valori di fiducia che A ha verso B.

Passando ad analizzare l'adattamento dei contenuti nei SASs deboli, analogamente ai forti, il passo del recupero di risorse è attuato con interrogazioni, questa volta però su folksonomie. Le queries basate sui tag possono essere processate con l'utilizzo di tecniche di disambiguazione, o mediante algoritmi di tag clustering, e in modo analogo alle ontologie, le query possono essere espanse o ristrette. Per quanto riguarda la seconda fase, il filtraggio dei contenuti, anche detto "content-based approach", i tag sono visti come parole chiave che raggruppati assieme descrivono una risorsa o un elemento. I profili degli utenti nei SASs deboli sono basati e costruiti in relazione ai tag che essi pongono, dunque vi sono vari approcci e tecniche che consentono di fare matching tra le caratteristiche e i concetti estratti dal profilo con i contenuti di varie risorse da proporre. Un primo approccio può essere l'uso di tecniche di ragionamento su folksonomie. Un esempio di sistema che lo

utilizza è Movie Recommender[20], un SAS che suggerisce e consiglia film ai vari utenti in base ai loro voti e i loro gusti. In questo sistema, similmente a quanto visto per CHIP nei SASs forti, i voti che gli utenti danno ad un determinato film vengono propagati alla sua “nuvola di tag”, o gruppo che lo descrive. In seguito il sistema applica determinate regole di ragionamento ed inferenza sui valori dei voti associati ai tag per computare la nuvola di tag personale associata ad ogni utente. Un altro approccio per il recupero di risorse consiste nel rappresentare sia il profilo utente sia le risorse con vettori di tag. In questo modo si possono così utilizzare tecniche di recupero d’informazioni basate su modelli di spazio vettoriale (ad esempio calcolando la distanza vettoriale tra il vettore dei tag degli interessi di un utente e i vettori dei tag di varie risorse). Quest’approccio tiene conto degli interessi di un utente in base ai tag che pone e utilizza, dunque ha il vantaggio di poter essere applicato, volendo, solo alle social annotations e non all’intero contenuto delle risorse. Altri approcci proposti sono gli “approcci collaborativi”(TBCF: Tag-Based Collaborative Filtering), in cui vengono considerati non solo i voti che i vari utenti esprimono sulle risorse, ma anche il loro modo di annotarle. In pratica in tali approcci il sistema calcola la distanza tra gli interessi dei vari utenti studiando i loro tag e cercando similarità tra essi (va a cercare utenti che hanno posto tag simili su di una stessa risorsa), in modo da raffinare gli algoritmi di filtraggio collaborativi. Mediante la similarità dei tag si può rilevare se utenti diversi hanno la stessa cognizione e le stesse ragioni d’interesse su di una data risorsa, dunque si aggiunge un nuovo parametro da considerare per la scelta dei contenuti da proporre, che è la vicinanza con utenti simili (con cognizioni simili a quelle dell’utente in esame).

Un altro approccio proposto, che si distacca un attimo dalle tecniche di recupero e filtraggio, consiste nella generazione automatica dei tag utente. Un sistema che utilizza tale approccio è P-TAG[23], in cui tag personalizzati per pagine Web sono generati automaticamente rispetto ai contenuti della pagina e a dati contenuti sullo user desktop. Tali tag possono poi essere utilizzati per vari scopi, tra cui in particolare ricerche Web personalizzate.

Infine l’ultimo approccio proposto per l’adattamento dei contenuti è quello riguardante i social network e modelli di fiducia tra utenti. Tale approccio è adottato principalmente da SASs ibridi, che utilizzano approcci misti, combinando strumenti e tecniche del Semantic Web con quelle del Web 2.0 (dunque tecniche semantiche forti con tecniche deboli). Un esempio significativo di sistema che adotta tale approccio è dato da Revyu, un sistema che permette di leggere e pubblicare recensioni, di votarle e suggerirle a ogni utente. Revyu è un sito Web 2.0 costruito con un infrastruttura del Semantic Web, in cui tutti i dati sono memorizzati in triple RDF, in particolare i recensori, e

dunque gli utenti, e le loro relazioni sociali sono descritti con FOAF. Usando questa infrastruttura vengono raccolte varie recensioni che vengono proposte agli utenti, i quali possono esprimere un voto. I voti raccolti da vari utenti su di una recensione vanno a determinare la validità di essa. Considerando invece i voti complessivi assegnati alle recensioni di un utente si costituisce il modello di fiducia dei vari utenti rispetto al singolo recensore. Ad esempio un utente che ha voti alti in tutte le recensioni da lui prodotte è sicuramente un utente a cui è riconosciuto un alto livello di fiducia nella comunità del social network, dunque le recensioni da lui prodotte sono tenute in maggior considerazione come contenuti da proporre e suggerire.

### **3.3.2 Adattamento della struttura e della presentazione**

Questo secondo tipo di adattamento consiste nell'utilizzo di tecniche semantiche per personalizzare la struttura di navigazione e le modalità di presentazione del sistema. Principalmente tali caratteristiche sono rappresentate da collegamenti (link) che forniscono una guida per gli utenti attraverso le risorse del sistema. Questo tipo di sistema produce l'effetto adattivo andando a gestire questi collegamenti. I link possono essere ordinati secondo vari criteri, visibili o invisibili, abilitati o disabilitati, e inoltre possono essere annotati in maniera personale con annotazioni adattive. Quest'ultima caratteristica va a costituire quello che è chiamato "adaptive navigation support", ovvero uno specifico gruppo di tecnologie che adattano il supporto alla navigazione dell'utente all'interno di un sistema tenendo conto dei suoi obiettivi, preferenze e conoscenze. Inizialmente questo gruppo di tecnologie furono sviluppate per gli Adaptive Hypermedia Systems, ma in seguito si è visto essere applicabili anche ad altri sistemi in ambito Web. Prima di passare a illustrare le varie tecniche semantiche forti e deboli utilizzate dai SASs, è bene precisare che in molti di essi, soprattutto i sistemi che producono raccomandazioni, i link sono talvolta parte dei contenuti proposti. In questi casi risulta dunque difficile distinguere tra adattamento dei contenuti e adattamento della struttura di navigazione, in quanto la demarcazione è minima e le due tipologie di adattamento spesso si sovrappongono.

Prima di illustrare gli approcci seguiti dai SASs forti e deboli, si vuole aprire un attimo una parentesi sugli appena citati Adaptive Hypermedia Systems. Questi ultimi sono una sorta di sistemi adattivi, tuttavia chiusi rispetto all'applicazione in campo Web. Lo scopo di tali sistemi è di guidare gli utenti ed evitare che si perdano nello "spazio degli hypermedia", ovvero aiutano nella selezione dei link e nella navigazione all'interno di un sistema, adattando la struttura di navigazione agli obiettivi, alle conoscenze e agli interessi di

ogni utente. Sono dunque sistemi usati principalmente per applicazioni di aiuto allo studio e all'apprendimento, ma sono in contrasto con le applicazioni e sistemi di e-Learning poiché non sfruttano risorse e materiali reperibili sul Web, e poiché non interoperano con altri sistemi. Infatti, come accennato a riguardo dell'integrazione e interoperabilità, il framework AHA! è stato sviluppato come supporto per tali sistemi, per garantire e fornire l'interoperabilità tra essi, e costruire dunque "Open Adaptive Hypermedia Systems". Questi sistemi raccolgono e scambiano le varie informazioni tra Adaptive Hypermedia Systems, e permettono a questi ultimi di operare in uno scenario più vasto.

Si presentano ora alcuni approcci utilizzati dai SASs forti. Dolog e Nejdl[2] hanno progettato un interessante approccio basato su un'ontologia per descrivere le strutture dei link, adattata da FOHM (Fundamental Open Hypertext Model) un modello per descrivere domini ipertestuali per gli Hypermedia Systems (ovvero permette di modellare spazi ipertestuali, navigazione all'interno di ipertesti e tassonomie ipertestuali). In particolare Dolog e Nejdl utilizzano tale ontologia per descrivere la struttura dei link di due SASs: PRF ed ELENA PLA, che come operazioni sui link attuano l'ordinamento e l'annotazione. ELENA PLA (Personal Learning Assistant) è un'applicazione e-Learning che aiuta un utente nell'apprendimento consigliando documenti da leggere, esercizi e formulando una sorta di piano di studio personalizzato in relazione agli obiettivi e alle conoscenze di ogni utente. L'approccio di Dolog e Nejdl usato in questi sistemi mette in luce, come sottolineano i loro autori, l'importanza di modellare propriamente i link e la loro struttura, al fine di generare percorsi personalizzati per ogni utente. Per fare ciò si rende necessaria l'ontologia per descrivere la loro struttura. Tale modello per descrivere le regole d'adattamento della struttura dei link, tuttavia, a differenza di quello formulato negli esempi di Dolog e Nejdl, può anche essere rappresentato in modo non semantico.

Un altro esempio di SAS forte che attua questo tipo di adattamento è HERA, un sistema che fornisce una metodologia per la generazione automatica di Web Information Systems (WIS). La metodologia descritta in HERA è di tipo model-driven e si possono distinguere tre parti nella fase di design: integrazione (infatti HERA è stato introdotto in precedenza parlando proprio delle problematiche dell'integrazione tra SASs), recupero dati, e generazione della presentazione. HERA fornisce una serie di funzionalità che consentono ad un WIS di gestire l'integrazione di nuove informazioni acquisite, e la generazione personalizzata della sua struttura presentazionale. HERA utilizza varie ontologie per rappresentare sia i concetti del dominio, che la struttura di navigazione, in termini di link e porzioni di contenuti. Diversamente le

regole e le condizioni che stabiliscono la visibilità dei link e l'apparenza delle porzioni di contenuti dipendono dalle caratteristiche dei dispositivi utilizzati e dell'utente (caratteristiche come ad esempio il fatto che un dispositivo utilizzato per interfacciarsi al sistema è un cellulare, un palmare, o un browser; oppure riguardo gli utenti per esempio, valutando il grado di esperienza e di conoscenza che l'utente ha su un determinato dominio, o su determinati concetti). Tali condizioni, associate ai link e ai frammenti di contenuti possono essere considerate regole d'adattamento, e sono rappresentate da proprietà RDFS associate ad essi. Un esempio per capire meglio questo concetto: quando un utente visualizza un certo frammento, o sceglie di visitare un determinato link, HERA acquisisce informazioni sul collegamento seguito dall'utente, e associa alle proprietà (definite in RDFS) legate a quel frammento o link, un valore maggiore. In questo modo tutti gli altri frammenti o link a cui sono associate le stesse proprietà verranno presentati dal sistema con un determinato colore, che indica all'utente che sono "link desiderabili", o consigliati. Applicando poi valori diversi, in base a vari parametri d'importanza il sistema va a definire vari colori da associare ai link o ai frammenti, a seconda del loro significato semantico, e al grado di rilievo per l'utente delle loro proprietà.

Per i SASs deboli, le tecnologie di Web 2.0 hanno introdotto un nuovo concetto di annotazioni adattive sociali (social adaptive annotations), da cui deriva di conseguenza il supporto ad annotazioni adattive sociali. L'effetto adattivo di tali supporti si basa questa volta sulle conoscenze collettive della comunità che fa uso del SAS. In particolare il nuovo concetto di "navigazione sociale" esprime l'idea che la navigazione dell'utente all'interno dello spazio dei link e delle risorse dipende ed è fortemente influenzata dai comportamenti degli altri utenti. Per esempio basti pensare all'azione di taggare: i tag posti da utenti su documenti o dati multimediali sono presi in considerazione da altri utenti, e li influenzano nell'andare a visitare quella data risorsa. Due esempi di SASs che utilizzano il supporto ad annotazioni adattive sociali sono AnnotatEd[70] e Knowledge Sea 2[71]. La particolarità di tali sistemi è che il supporto alle annotazioni non è limitato solamente al taggare e a prendere visione di altre annotazioni, ma dal fatto che essi sono in grado di produrre "meta-annotazioni". AnnotatEd è un SAS che consente il browsing e la ricerca di risorse educative sul Web. Questo SAS, prima di mostrare il contenuto delle pagine Web selezionate all'utente, annota i link selezionati con due tipi di annotazioni: derivanti da tecniche tradizionali, o di tipo sociale, ovvero basate su tag posti da utenti specifici e da gruppi di utenti che hanno preso visione di quella pagina Web proposta. Knowledge Sea 2 è un sistema analogo ad AnnotatEd, ovvero che fornisce una sorta di mappa, o piano di naviga-

zione attraverso risorse Web per supportare l'educazione e l'apprendimento degli utenti.

### 3.3.3 Adattamento per supportare le annotazioni

Quest'ultima tipologia di adattamento è mirata a supportare gli utenti nell'annotazione delle risorse, aiutandoli a porre tag su esse. E' chiaro che tale adattamento riguarda in gran parte i SASs deboli, in particolare i sistemi con un alto impatto sociale, come i social network, ed è stato introdotto per meglio classificare il tipo di tagging sociale, al fine di raccomandare tag appropriati agli utenti. Tuttavia questo tipo di adattamento trova anche riscontro nella semantica forte, poiché può fornire il supporto per lo sviluppo di ontologie personali.

L'unico esempio (che si può trovare nella letteratura) di sistema che realizza un adattamento per il supporto di annotazioni per SASs forti è quello di Bibster[69], un semantic tool peer-to-peer per la gestione e la condivisione di bibliografie. Bibster classifica le sue risorse basandosi su un'ontologia OWL che ne descrive il dominio. L'adattamento è dato dall'utilizzo di un algoritmo collaborativo di filtering, per aiutare gli utenti a gestire l'evoluzione e i cambiamenti della loro ontologia personale, suggerendo loro possibili cambi di dominio all'interno dell'ontologia, durante le loro attività. L'obiettivo è di determinare l'importanza e la rilevanza dei cambiamenti nelle ontologie personali, basandosi per esempio sulla similarità di cambiamenti fatti da altri utenti sulla propria ontologia.

Riguardo il supporto alle annotazioni ci sono più lavori che riguardano i SASs deboli. Gli obiettivi dell'adattamento per il supporto di annotazioni, per essi sono:

- suggerire agli utenti tag appropriati da porre sulle risorse: questo tenendo in considerazione sia gli utenti che le risorse e le loro caratteristiche;
- stimolare e facilitare gli utenti nel porre annotazioni (stimolarli a porre tag);
- migliorare il recupero di una risorsa: questo sia per l'utente che annota quella determinata risorsa, che per tutti gli altri utenti;
- rendere i tag "più semantici".

Per raggiungere questi obiettivi, e attuare l'adattamento per il supporto di annotazioni, si possono utilizzare gli stessi approcci illustrati per l'adattamento dei contenuti, applicandoli però ai tag (invece che utilizzarli per suggerire

contenuti, sono usati per suggerire tag). Dunque per la raccomandazione di tag si possono utilizzare i “content-based approach” e gli approcci collaborativi. Più in dettaglio la tag recommendation di tali approcci si basa prendendo in considerazione e analizzando le seguenti informazioni:

Approcci collaborativi:

- tag posti da altri utenti sulla stessa risorsa, o su risorse simili;
- tag posti da utenti simili.

Approcci basati sui contenuti:

- tag posti dallo stesso utente su risorse simili tra loro;
- caratteristiche delle risorse.

Partendo dagli approcci collaborativi, un primo SAS da prendere in considerazione come esempio è AutoTag[24], un tool che suggerisce tag per post dei Weblog, utilizzando metodi di filtraggio collaborativi. Più precisamente, i tag sono suggeriti in AutoTag esaminando quelli già assegnati a post simili (quindi analizzando risorse simili a quella da annotare). Per rilevare questi ultimi vengono utilizzate metriche derivanti dal recupero di informazioni basate su un indicizzazione delle risorse. Un secondo esempio, che è stato parzialmente implementato in Yahoo! My Web 2.0[72], s’identifica questa volta nel suggerire i tag in base ad altri posti da altri utenti sulla stessa risorsa. In Yahoo! sono definiti set di “criteri di buon tagging”, e tali criteri vengono poi dati ad un algoritmo che raffina la scelta dei tag da proporre tra quelli posti dagli altri utenti su quella risorsa. Tali criteri di raffinamento sono utili giacché coprono vari aspetti, come fornire una rappresentazione multi faccettata di una risorsa, mostrare la popolarità dei tag e ridurre lo spam. Un ultimo approccio collaborativo da analizzare, riguarda il suggerimento di tag in base alla similarità di utenti (dunque tag posti da utenti simili). Un approccio proposto è l’uso di tradizionali tecniche di filtri collaborativi per trovare utenti simili. Tali tecniche fanno uso di algoritmi che calcolano la distanza tra utenti a partire da tag o risorse taggate comuni tra gli utenti in esame. Un esempio è dato da FolkRank[25], un algoritmo che sfrutta la struttura delle folksonomie, e dunque i tag e i voti degli utenti, e le risorse taggate. FolkRank può essere utilizzato per due scopi: determinare una valutazione, e visione generale su di una data risorsa da parte della comunità di utenti; oppure per ottenere varie valutazioni più dettagliate su essa, da diversi gruppi più specifici di utenti (e dunque identifica gruppi di utenti simili).

Gli approcci basati sui contenuti invece suggeriscono tag comparando i contenuti delle risorse e i profili degli utenti. Un esempio di tale approccio si vede in Smart Tag Recommender[26], un SAS che similmente ad AutoTag suggerisce tag all'utente per risorse e documenti. Smart Tag consiglia in base ai tag che l'utente ha già posto su altre risorse, e in base a caratteristiche dei loro contenuti. Più precisamente va a cercare altri documenti simili da cui prendere e analizzare i tag, per scegliere quelli da proporre. Un'altra particolarità di questo sistema è che considera la storia dei tag posti dall'utente, memorizzata nel suo profilo. In pratica, per ogni tag che l'utente pone, all'interno del suo profilo viene costruito un classificatore di Bayes<sup>3</sup>. Ogni volta che c'è bisogno di suggerire un tag all'utente per un dato documento, quest'ultimo viene processato (vengono cercati altri documenti con caratteristiche di contenuto simili da cui analizzare i tag) e i tag estrapolati vengono matchati e processati con i classificatori del profilo utente. Ognuno di questi classificatori predice in modo probabilistico quale dovrebbe essere il tag che meglio si addice, e che l'utente dovrebbe associare al documento in esame.

In base agli obiettivi definiti all'inizio di questa sezione, gli approcci finora illustrati sono mirati principalmente al miglioramento del recupero di risorse, al suggerimento e stimolazione nel tagging per migliorare la qualità delle folksonomie. Manca da vedere qualche approccio per rendere i tag "più semantici". Innanzitutto è bene capire l'utilità di quest'obiettivo. La tag recommendation per aumentare la semantica dei tag è utile per tecniche di reasoning, al fine di andare a produrre e costruire ontologie in modo collaborativo. Tali approcci sono per l'appunto chiamati "ontology-based approaches". La maggior parte di essi sono approcci misti, ovvero utilizzano sia tecniche di Semantic Web che di Web 2.0. Con le prime i sistemi acquisiscono suggerimenti di tag sulle risorse, e con le seconde viene messa a disposizione dell'utente la possibilità di accettare o rifiutare i tag proposti. Successivamente viene attuato un processo di costruzione collaborativo, il cui risultato può essere una rigorosa ontologia, un'ontologia "rilassata" oppure può convergere in una folksonomia. Un esempio di questo tipo di approcci si trova in Matching Component[1], un modulo che compara set di tag posti dagli utenti con set di ontologie o con altri insiemi di tag che sono già stati precedentemente accettati nella conoscenza di base. In seguito il sistema suggerisce all'utente nuovi tag alternativi

---

<sup>3</sup>Classificatore di Bayes: è un classificatore probabilistico basato sull'applicazione del teorema di Bayes (facendo uso di forti assunzioni di indipendenza: "classificatore bayesiano naïf"). Esso è usato per determinare in modo probabilistico l'appartenenza di un'entità ad una data classe, in base alla presenza, o assenza di determinate caratteristiche. Ad esempio un frutto può essere considerato una mela se è rosso, tondo e ha un diametro di circa quattro centimetri; considerando queste e altre caratteristiche che un determinato frutto può avere o non avere, è possibile stabilire in modo probabilistico se esso è una mela oppure no (la probabilità che quel frutto sia una mela).



e quest'ultimo può decidere se accettarli oppure no. In base alla sua scelta, e alle scelte precedenti (dunque alle accettazioni/rifiuti di tag precedenti) il sistema costruisce un'ontologia rilassata. In breve il sistema tenta di attribuire lui stesso un significato semantico ai tag, basandosi su ontologie esistenti o tag il cui significato semantico è già stato accettato, e usa l'utente per verificare se l'attribuzione semantica che ha fatto è corretta. Un altro esempio è CommonFolks[27], un sistema che propone un metodo collaborativo per annotare oggetti di apprendimento, cercando di evitare inconsistenze nelle annotazioni. Le annotazioni in CommonFolks sono definite come statements RDF, e i tag posti dagli utenti sono considerati istanze di WordNet[73] (una sorta di dizionario in cui i termini sono suddivisi e classificati in thesauri, e a cui sono attribuiti significati semantici) se già presenti in esso, altrimenti sono considerati come nuove istanze da aggiungervi. Questa aggiunta viene fatta tramite un tool che consiglia la posizione del tag all'interno dell'ontologia di WordNet, sfruttando i feedback dati dalle risposte ai suggerimenti sulla posizione del tag, provenienti dalla maggior parte degli altri utenti. Tali approcci mettono in luce i validi risultati che si possono ottenere dall'integrazione di ontologie e folksonomie. Tuttavia, per raggiungere buoni risultati e ottenere rigorose ontologie, è necessario un grande sforzo, acquisendo feedback da parte di tanti utenti. Quest'ultimo fattore costituisce una limitazione all'efficacia di tali approcci. Per esempio risulta difficile acquisire tanti feedback, e consolidare quindi un'ontologia in SASs che restano attivi solo per un breve periodo, o che possiedono uno scarso numero di utenti, e dunque sono poco utilizzati.

## Capitolo 4

### “Pillo SAS’s”: la realizzazione di un SAS

Come ultimo punto di questa tesi, si illustra un esempio di SAS realizzato dall'autore, allo scopo di comprendere meglio questi sistemi utilizzando le tecniche semantiche illustrate, e gli strumenti messi a disposizione dal Semantic Web. Pillo SAS's[74] è un sistema adattivo semantico costruito su di un'infrastruttura Web. E' una sorta di sito Web adattivo, in cui gli utenti possono accedere e leggere vari articoli di alcune categorie: sport, ambiente, automobili e cucina. Il sistema registra gli interessi di ogni utente e in base a essi consiglia e suggerisce articoli correlati, appartenenti alle stesse categorie o ad altre correlate o vicine a esse. Pillo SAS's è dunque un sistema che a livello di funzionalità si presenta simile ad altri sistemi illustrati in questa tesi, come ad esempio Revyu o Movie Recommender, siti Web, o Weblogs che consigliano agli utenti recensioni di vario genere. Prima di mostrare nel dettaglio le tecniche utilizzate e spiegarne brevemente il funzionamento è bene classificare questo sistema e definire il tipo di processo adattivo che adotta. Pillo SAS's può essere considerato un SAS forte, in quanto fa uso di tecniche semantiche forti, di ontologie note, come FOAF, Dublin Core, e un'altra (che in seguito verrà illustrata brevemente) usata per descrivere una classificazione dei tag e delle rispettive categorie. Un altro strumento dei sistemi forti utilizzato da Pillo SAS's è SPARQL, come linguaggio di interrogazione per estrarre dalle ontologie concetti e contenuti. Per quanto riguarda invece il tipo di processo adattivo, Pillo SAS's attua principalmente un adattamento dei contenuti, che in parte può essere anche visto come un adattamento della struttura di navigazione (si ricorda che il divario tra questi due tipi di adattamento non è sempre netto e ben definito). Ora si descrive il funzionamento in

modo completo di Pillo SAS's seguendo il modello di funzionamento generale dei SASs definito (figura 4.1).

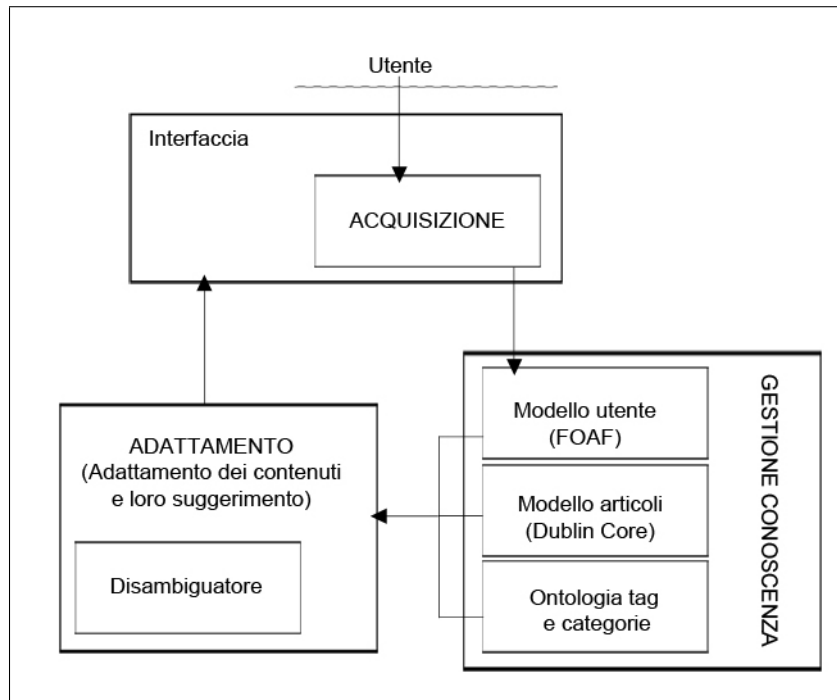


Figura 4.1: Schema riassuntivo del funzionamento di Pillo SAS's.

## 4.1 Acquisizione

In Pillo SAS's le informazioni relative ai documenti e agli articoli che il sistema propone ai suoi utenti sono già presenti, e la conoscenza relativa ad essi è raccolta e mantenuta aggiornata dal creatore del sistema. Le informazioni a riguardo degli utenti sono invece acquisite dall'esterno dinamicamente. Tali informazioni riguardano alcuni dati sull'utente, acquisiti in fase di registrazione al sito, e informazioni sui suoi interessi, acquisite dal sistema progressivamente durante la navigazione dell'utente tra gli articoli messi a disposizione. L'idea su cui si basa l'acquisizione di questo tipo di conoscenza è molto simile a quella espressa per i SASs deboli a proposito delle azioni di tagging: come un utente di solito pone dei tag su una risorsa a cui è interessato, analogamente in Pillo SAS's un utente va a visualizzare un articolo a cui è interessato. Il fatto che un utente scelga di prendere visione di un articolo, cliccando sul rispettivo link, è considerato come un interesse che quell'utente

ha rispetto a esso. Per quanto riguarda lo scambio di dati tra SASs, c'è da dire che Pillo SAS's non presenta problematiche relative a questo aspetto, in quanto non collabora e non coopera con altri sistemi, dunque lo scambio di dati è inesistente.

## 4.2 Gestione della conoscenza

Pillo SAS's utilizza e gestisce vari modelli che si riferiscono a differenti tipi di conoscenza. Partendo dal dominio troviamo due modelli principali: un primo documento RDF organizza i tag in una gerarchia di categorie, e quindi si può considerare una sorta di ontologia. In questo documento è stata modellata la conoscenza relativa a categorie di tag presenti negli articoli del sistema. Per chiarire meglio, gli articoli messi a disposizione sono descritti mediante lo schema di Dublin Core, e in particolare nelle loro descrizioni sono presenti dei tag che specificano i soggetti e i contenuti principali di ogni documento (i tag della proprietà `subject`). Affinché il sistema possa trovare articoli simili da proporre ai suoi utenti è necessaria una descrizione semantica di tali tag, e una loro attribuzione a una qualche categoria. Ed è dunque questo il compito di quest'ontologia, cioè quello di descrivere in modo semantico, anche se molto semplice (sono definite in sostanza solo alcune categorie e sottocategorie da associare ai tag), il dominio di conoscenza dei tag. Ad esempio per capire meglio il compito di tale ontologia, al tag "gatto" è attribuita la categoria `felini`, la quale è sottocategoria di `animali`, che a sua volta sta in ambiente. In particolare in Pillo SAS's sono modellati dei domini "minimali" (non in modo molto vasto e dettagliato) relativi ad automobili, sport, ambiente e cucina. Quest'ontologia si può considerare simile a WordNet o alla menzionata ontologia di Wikipedia, in cui sono gestiti vocaboli e categorie di parole. Ovviamente quella costruita in questo sistema è molto più piccola e contiene un numero ristretto e poco dettagliato di concetti. Questo poiché, com'è stato detto, le tecnologie del Semantic Web sono sì efficaci, ma possono essere molto costose, infatti per modellare e descrivere domini molto vasti e ampi è richiesto molto tempo e un grande sforzo. Il secondo modello utilizzato è quello cui si è già accennato, un documento RDF che utilizza lo schema Dublin Core, in cui sono descritti questa volta gli articoli e alcuni metadati relativi a essi. In aggiunta in questo modello, alle proprietà di Dublin Core, vi è anche una proprietà "category" utilizzata per fare riferimento a una delle categorie base di Pillo SAS's (e dunque sport, automobili, ambiente e cucina), il cui scopo sarà spiegato dettagliatamente in seguito. Queste sono le informazioni che riguardano la modellazione dei vari domini trattati nel sistema (dominio degli

articoli, e dominio dei tag). Come menzionato nella fase di acquisizione, tali informazioni sono già presenti all'interno del sistema. I metadati relativi agli articoli sono posti dagli autori, come la classificazione dei tag è fatta da chi la produce e descrive, quindi tali informazioni non vengono acquisite esternamente come può essere per le folksonomie. Quest'ultima caratteristica, come descritto all'inizio di questo lavoro (nella distinzione tra Semantic Web e Web 2.0) aiuta ad identificare Pillo SAS's come un SAS forte. Passando ora alla gestione del profilo utente, i dati e le informazioni acquisite dagli utenti sono salvati in documenti FOAF. In particolare nel file FOAF che modella le conoscenze relative ad ogni utente sono di volta in volta aggiornati gli interessi. Tutte le volte che un utente decide di visitare un articolo nuovo, questo viene aggiunto alla lista dei suoi interessi nel suo profilo. In tal modo questo tipo di conoscenza è continuamente estesa dal sistema aggiungendo informazioni che gli provengono dall'interazione con gli utenti. Ricordando quanto detto poco fa, Pillo SAS's non gestisce l'interoperabilità con altri sistemi, dunque non acquisisce informazioni da altri SASs. Tuttavia grazie l'utilizzo di FOAF e Dublin Core, e grazie alla grande flessibilità del Resource Description Framework, lo scambio di informazioni con altri sistemi che utilizzano queste infrastrutture semantiche si potrebbe realizzare in modo agevole. Anche per quanto riguarda l'estensione per mezzo di ragionamenti, non sono state introdotte regole d'inferenza sulle ontologie, questo poiché i concetti utilizzati in esse sono di per sé semplici ed estrapolabili semplicemente mediante l'utilizzo di query.

## 4.3 Processo d'adattamento

Il tipo di adattamento attuato si può identificare con un adattamento dei contenuti ed un loro suggerimento. Infatti, il sistema suggerisce articoli simili a quelli visitati dall'utente, o comunque associati alle stesse categorie di questi articoli o ad altre vicine a esse. Si trova dunque un riscontro con gli approcci semantici proposti per questo tipo di adattamento. In particolare si fa uso dell'approccio basato sul recupero di risorse e concetti, e filtraggio delle informazioni. In primo luogo il sistema va a reperire le categorie a cui è interessato l'utente in questione, andando ad interrogare con query SPARQL il suo file FOAF, recuperando i suoi interessi (in particolare recupera gli uri, i riferimenti alle risorse/articoli a cui è interessato). A ogni articolo disponibile nel sistema sono associate una serie di categorie tra quelle presenti nella rispettiva ontologia. Tale associazione è fatta interrogando (sempre con query in SPARQL) prima il documento Dublin Core di ogni articolo, per

andare a prendere le sue informazioni, in particolare i tag del subject, e la sua categoria base di riferimento (o categoria madre, descritta dalla proprietà “category”). In seguito passando i tag del subject come parametri alle query fatte sull’ontologia delle categorie, si estraggono le eventuali categorie e sopracategorie associate. Ricapitolando, dai documenti si estraggono i tag, e dai tag si estraggono le categorie a cui appartengono. A questo punto, nella descrizione dell’adattamento dei contenuti è stato menzionato l’eventuale uso di tecniche di ragionamento: in questo caso è utilizzata una tecnica che si può considerare di “ragionamento”, anche se non è costruita con regole semantiche per ontologie. Con questa tecnica, le categorie estrapolate vengono “pulite” da un modulo chiamato “disambiguatore”. Lo scopo di quest’ultimo è quello di eliminare dalla lista di categorie ottenute eventuali categorie di appartenenza ambigua, o per meglio dire “di disturbo”. Un esempio per capire meglio questo aspetto potrebbe essere il seguente: il sistema mette a disposizione dei suoi utenti tre articoli taggati col tag “Ferrari”, dove uno riguarda la Ferrari intesa come l’automobile, il secondo è riferito alla ginnasta Vanessa Ferrari, ed il terzo fa riferimento ad una nota marca di spumante italiano. Il tag Ferrari risulta dunque avere più di un significato. Ora, interrogando l’ontologia (delle categorie) col tag Ferrari sono proposte categorie relative sia ad automobili, che a sport e a cucina, e il sistema non è in grado di capire precisamente a quale di queste tre categorie si riferisce il tag. Il disambiguatore serve proprio per risolvere questo problema. Per funzionare sfrutta il tag category aggiunto alle descrizioni Dublin Core di ogni articolo, e tale tag ha il compito di identificare la categoria madre cui l’articolo si riferisce. In questo modo con la categoria madre può interrogare di nuovo l’ontologia delle categorie, ed estrarre tutte e sole le sue sottocategorie, che saranno le uniche da considerare per quel dato articolo. Il disambiguatore elimina così dalla lista di categorie rilevate, tutte quelle che non appartengono alla categoria madre dell’articolo che sta pulendo, disambiguando i molteplici significati e riferimenti di uno o più tag condivisi in più categorie. Il passo successivo nel processo di adattamento è il filtraggio delle informazioni, per andare a scegliere i contenuti da proporre, e quindi gli articoli. In questa fase, dalla lista d’interessi dell’utente il sistema recupera le informazioni riguardanti gli articoli visitati, con le rispettive liste di categorie associate già pulite (o disambiguate). A questo punto per ogni articolo va a cercarne altri che abbiano associate le stesse categorie, o categorie vicine ad esse (dove per queste ultime si intendono le eventuali sopracategorie). In particolare a ognuna di esse sono associati vari livelli di specificità, e Pillo SAS’s attua la sua ricerca prima a un livello più specifico, e nel caso non trovi niente per esso, passa a cercare ai livelli superiori. Ad esempio, per un articolo riguardante Igor Cassina, famoso ginnasta italiano, il sistema troverà associate le categorie ginnasti (che

è la più specifica), ginnastica artistica, sport (meno specifica). Come prima cosa cercherà altri articoli della categoria ginnasti, e se li trova li usa come suggerimenti verso l'utente. Nel caso non trovi altri articoli che si riferiscono a ginnasti, ne cercherà altri della categoria ginnastica artistica, e se non trova niente neanche per essi, proporrà quelli più in generale che hanno associata la categoria sport. Questo criterio di selezione rappresenta il filtering dei contenuti, che nel caso di Pillo SAS's sono gli articoli da suggerire.



Figura 4.2: Screenshot di Pillo SAS's: l'utente non ha ancora letto alcun articolo, ma ha scelto di visualizzare l'elenco di quelli della categoria "Cucina".

L'altra tipologia di adattamento che si può intravedere in questo sistema, è l'adattamento della struttura di navigazione e della presentazione dei contenuti, che tuttavia si sovrappone con l'adattamento stesso dei contenuti. Gli articoli suggeriti agli utenti sono infatti presentati come link, con una breve descrizione della pagina Web in cui è scritto l'articolo. Il sistema altro non fa che presentare/abilitare alcuni link rispetto ad altri, e quindi creare una sorta di aiuto, o guida personalizzata per la navigazione all'interno degli articoli presenti sul sito. In ogni modo è più considerevole l'aspetto adattivo dei contenuti, rispetto a quest'ultimo, che si può considerare come secondario. Nelle figure 4.2, 4.3, 4.4 sono presentati alcuni screenshot che mostrano un esempio concreto di utilizzo di Pillo SAS's.

Per concludere questa parte si possono fare delle brevi considerazioni su questo lavoro: anche se il sistema sviluppato non gestisce l'interoperabilità e lo



Figura 4.3: Screenshot di Pillo SAS's: l'utente ha scelto di leggere un articolo sugli spumanti, e il sistema gli ha suggerito altri articoli che riguardano vini.

scambio di informazioni e dati con altri SASs, grazie alla rappresentazione dei dati mediante statements RDF si potrebbe gestire in modo semplice questo aspetto, ad esempio grazie all'utilizzo di qualcuno degli approcci illustrati in questo lavoro. Una seconda considerazione si può fare a riguardo degli articoli e della modellazione del dominio delle varie categorie. Il numero di articoli messi a disposizione da Pillo SAS's è ristretto, come lo è anche il numero di categorie e sottocategorie modellate e gestite per i tag. Questo è giustificato dal fatto che descrivere domini molto grandi e ampi è difficile e richiede molto tempo e sforzo. Per questo motivo la modellazione fatta in Pillo SAS's è stata minima, ai fini didattici e dimostrativi, mettendo a disposizione solo quattro categorie madre (sport, automobili, cucina e ambiente), e alcune loro sottocategorie. Un'altra motivazione viene dal fatto che descrivere domini molto ampi e complessi rende anche più complessa la loro comprensione e l'elaborazione, da un punto di vista progettuale, di casi di test per il sistema. Per tale motivo si è cercato di contenere il lavoro in un contesto abbastanza semplice. Infine va ricordato che questo lavoro non è incentrato principalmente sulla modellazione, ma sull'aspetto di adattamento dei sistemi. Nonostante questo, i modelli utilizzati possono comunque essere ampliati, e meglio descritti e dettagliati, per espandere i vari domini sulle categorie dei tag, e per migliorare il livello semantico di conoscenza del sistema. Come nota personale dell'autore, questo lavoro è stato molto utile, oltre che





Figura 4.4: Screenshot di Pillo SAS's: l'utente ha selezionato un articolo sul leone. L'articolo è stato aggiunto alla lista dei visitati, e il sistema ha restituito tra i suggerimenti altri articoli riguardanti i felini.

per capire meglio alcuni concetti studiati relativi ai SASs, anche come prima esperienza nell'utilizzo di tecnologie semantiche, e come ulteriore esperienza nell'utilizzo di tecnologie Web.

# Conclusioni

Al termine di quest'analisi sui Sistemi Adattivi Semantici, si riassumono i concetti principali visti. Una volta introdotte le tecnologie del Semantic Web e del Web 2.0, mettendo in luce le loro principali caratteristiche, è stata fatta una classificazione dei SASs relativamente ad esse. Vi sono SASs forti, che grazie agli strumenti offerti dal Semantic Web fanno uso di tecniche semantiche forti. Questi SASs utilizzano dunque ontologie, linguaggi d'interrogazione, e linguaggi per esprimere ragionamenti su esse. Vi sono poi SASs deboli, più vicini al mondo del Web 2.0, che utilizzano tecniche semantiche deboli. In questa categoria si collocano anche quelli che utilizzano approcci ibridi, o misti (dunque sia semantica forte, come ontologie, che Web 2.0, come folksonomie). In genere tali tecniche deboli fanno uso di folksonomie, utili per modellare domini instabili, e validi strumenti per il recupero e la classificazione di risorse.

Una volta presentate queste tipologie di SASs sono stati analizzati alcuni strumenti e tecnologie utilizzate da essi per supportare il processo di adattamento con varie tecniche e approcci. Queste tecniche e approcci sono stati studiati, mostrando anche alcuni esempi di loro applicazioni, sempre tentando di mantenere una certa distinzione tra SASs forti e SASs deboli, anche se talvolta alcune tecniche forti e deboli possono sembrare simili tra loro. Tali tecniche sono risultate utili per illustrare alcune delle modalità di funzionamento dei SASs, e per risolvere varie problematiche legate alle loro componenti principali (secondo il modello generale proposto) che trattano:

- l'acquisizione di dati e informazioni;
- la gestione della conoscenza;
- l'attuazione del processo d'adattamento.

Dall'analisi emergono alcune considerazioni che vale la pena di esplicitare:

- I SASs in generale risultano sistemi utili al giorno d'oggi per personalizzare vari servizi e semplificarne l'utilizzo da parte degli utenti. Essi

inoltre si possono considerare come dei piccoli agenti intelligenti, poiché sono in grado di comprendere diversi tipi di dati, e di sfruttarli per personalizzare i loro servizi. Sono sistemi ancora “giovani”, nati e sviluppati negli ultimi anni, che quindi non sono ancora stati adottati ampiamente in molti campi. I SASs devono ancora crescere e migliorare, e sicuramente con la loro crescita si crede che troveranno un maggiore utilizzo.

- I SASs forti trovano applicazione soprattutto nel campo dell’e-Learning e in sistemi come Weblogs, in cui sono presenti molte risorse e contenuti. In particolare questi sistemi sono mirati principalmente al suggerimento di contenuti, o alla creazione di guide e percorsi personalizzati all’interno di essi. Emerge dunque che le tipologie di adattamento più comunemente attuate dai SASs forti sono l’adattamento dei contenuti e loro suggerimento, e l’adattamento della struttura di navigazione e della presentazione.
- Sempre relativamente ai SASs forti si può fare una considerazione sull’uso delle ontologie per rappresentare vari tipi di conoscenza. Esse risultano essere uno strumento molto potente, che può essere utilizzato da varie tecniche (interrogazioni, ragionamenti, ecc.), tuttavia per rappresentare una conoscenza ampia e completa su determinati domini, magari molto vasti, può servire un grande sforzo e molto tempo per codificare tutti i concetti e le relazioni tra essi. Le ontologie sono quindi strumenti che possono presentare problemi di applicabilità a breve termine per questi sistemi.
- I SASs deboli invece trovano una maggiore applicazione nei social network. Dato che i social network si stanno diffondendo ed evolvendo sempre di più ai giorni nostri, indubbiamente anche i SASs deboli si diffonderanno ed evolveranno con loro.
- I SASs deboli si presentano meno potenti, rispetto ai SASs forti, nella rappresentazione di tipi diversi di conoscenza. Tuttavia grazie all’uso di folksonomie sono in grado di gestire domini instabili, non ancora ben definiti, ed in evoluzione. Principalmente emerge che la conoscenza a cui sono più legati e che sfruttano di più è quella relativa agli utenti. In particolare i SASs deboli tengono fortemente in considerazione i tag e le social annotation che gli utenti generano e utilizzano, e cercano di studiare il loro modo di porre questi tag.
- A riguardo dei SASs ibridi si può dire che mostrano alcuni interessanti approcci di come combinare l’uso di ontologie e folksonomie, che possono produrre validi risultati. Tuttavia appare evidente che molti di

questi approcci presentano delle limitazioni e sono ancora in una fase sperimentale, devono ancora evolversi e migliorare.

- La tecnica di adattamento per il supporto ad annotazioni risulta essere utilizzata prevalentemente dai SASs deboli e da quelli ibridi.
- Un ultima osservazione si può fare sulle tecniche semantiche osservate, che per i SASs (in generale) sono risultate molto utili ed efficaci per risolvere problematiche legate al loro funzionamento, ma questa è solo una delle loro tante applicazioni. Credo infatti che esse possano essere utilizzate per molti altri scopi, e da altri sistemi al di là di quelli adattivi.

Infine è stato descritto, seguendo il modello funzionale esposto in precedenza, un esempio completo di funzionamento e realizzazione di un SAS forte, il sistema “Pillo SAS’s”, realizzato come progetto dall’autore a fini didattici e di apprendimento. La funzionalità di questo sistema consiste nel mostrare e suggerire articoli ai suoi utenti in base ai loro interessi, attuando dunque un adattamento dei contenuti. Pillo SAS’s si può considerare un SAS forte poiché sfrutta le ontologie FOAF e Dublin Core per descrivere vari tipi di conoscenza, sia acquisita, che già in suo possesso. Inoltre definisce utilizzando RDF un suo, seppur semplice, sistema di classificazione di tag e categorie per descrivere gli articoli presenti nel sistema, che si può considerare come una piccola ontologia. Costruire rigorose ontologie è un processo lungo e per domini molto vasti richiede molto tempo. Per questo motivo l’ontologia di Pillo SAS’s non è molto vasta e dettagliata, e di conseguenza anche il numero di articoli messi a disposizione dal sistema è limitato. Considerando questi ultimi aspetti, alcuni possibili sviluppi per Pillo SAS’s potrebbero consistere nel:

- ampliare il numero di articoli presenti nel sistema;
- ampliare la gerarchia di categorie e tag dell’ontologia, espandendo lo spazio dei tag di ogni categoria specifica, e aggiungendo nuove categorie intermedie e categorie madre;
- farlo interoperare con qualche altro SAS, consentendogli magari di scambiare informazioni sugli articoli e sugli utenti, per ampliare la sua conoscenza.

# Bibliografia

- [1] Ilaria Torre - *Adaptive systems in the era of semantic and social web, a survey*. (2009)
- [2] Peter Dolog and Wolfgang Nejdl - *Semantic Web Technologies for the Adaptive Web*. (2007)
- [3] Paul De Bra e Licia Calvi - *AHA! An open Adaptive Hypermedia Architecture*.
- [4] M.T. Pazienza - *Metodologie per la gestione di conoscenza ontologica*. (2008-2009)
- [5] Jacopo Farina - *Assegnamento automatico di macrocategorie agli articoli di Wikipedia*. (2009-2010)
- [6] Marieke Guy e Emma Tonkin - *Folksonomies - Tidying up Tags?*. (2006)
- [7] Tom Heath and Enrico Motta - *How to Combine the Best of Web2.0 and a Semantic Web: Examples from Revyu.com*. (2007)
- [8] Aroyo, Dolog, Jan Houben, Kravcik, Wild, Naeve e Nilsson - *Interoperability in Personalized Adaptive Learning*. (2006)
- [9] Akkermans, Aroyo e Bellekens - *iFanzzy: Personalised Filtering Using Semantically Enriched TVAnytime Content*. (2006)
- [10] Nicola Henze - *Personalization Services for the Semantic Web: The Personal Reader Framework*.
- [11] Peter Brusilovsky - *Adaptive Navigation Support: From Adaptive Hypermedia to the Adaptive Web and Beyond*. (2004)
- [12] Fabian Abel, Mischa Frank, Nicola Henze, Daniel Krause e Patrick Siehnde - *GroupMe! – Combining Ideas of Wikis, Social Bookmarking, and Blogging*.
- [13] Mark van Setten, Stanislav Pokraev, Johan Koolwaaij - *Context-Aware Recommendations in the Mobile Tourist Application COMPASS*.

- [14] Carmagnola e Dimitrova - *An Evidence-Based Approach to handle Semantic Heterogeneity in Interoperable Distributed User Model*. (2008)
- [15] Carmagnola et al. - *iCITY - an adaptive social mobile guide for cultural events*.
- [16] R.Kernchen, David Bonnefoy, A. Battestini, B. Mrohs, M. Wagner e M. Klemettinen - *Context-Awareness in MobiLife*.
- [17] Shlomo Berkovsky, Tsvi Kuflik, Francesco Ricci - *Mediation of user models for enhanced personalization in recommender systems*. (2008)
- [18] Lora Aroyo, Riichiro Mizoguchi e Christian Tzolov - *OntoAIMS: Ontological Approach to Courseware Authoring*.
- [19] William T. Niu, Judy Kay - *PERSONAF: framework for personalised ontological reasoning in pervasive computing*. (2010)
- [20] Martin Szomszor, Ciro Cattuto, Harith Alani, Kieron O'Hara, Andrea Baldassarri, Vittorio Loreto, Vito D.P. Servedio - *Folksonomies, the Semantic Web, and Movie Recommendation*. (2007)
- [21] Peter Dolog - *An approach to personalization in the Elena project*. (2003)
- [22] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, Margeritta von Wilamowitz-Moellendorff - *Gumo - The General User Model Ontology*. In *Proceedings of User Modeling*. (2005)
- [23] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, Siegfried Handschuh - *P-TAG: large scale automatic generation of personalized annotation tags for the web*. (2007)
- [24] Gilad Mishne - *AutoTag: a collaborative approach to automated tag assignment for weblog posts*. (2006)
- [25] Andreas Hotho, Robert Jaschke, Christoph Schmitz, Gerd Stumme - *FolkRank: A Ranking Algorithm for Folksonomies*.
- [26] Pierpaolo Basile, Domenico Gendarmi, Filippo Lanubile, Giovanni Semeraro - *Recommending Smart Tags in a Social Bookmarking System*. (2007)
- [27] Scott Bateman - *How can we describe and use digital learning materials more effectively?* (2006)
- [28] <http://www.w3.org/2001/sw/wiki/RDF> - *Resource Description Framework (RDF)*. (10 febbraio 2004)

- [29] <http://www.w3.org/TR/REC-rdf-syntax/> - *RDF/XML Syntax Specification (Revised)*. (W3C Recommendation 10 febbraio 2004)
- [30] <http://www.w3.org/TR/rdf-schema/> - *RDF Vocabulary Description Language 1.0: RDF Schema*. (W3C Recommendation 10 febbraio 2004)
- [31] <http://xml.html.it/guide/lezione/1759/rdf/> - *RDF*.
- [32] <http://www.w3.org/TR/owl-features/> - *OWL Web Ontology Language*. (W3C Recommendation 10 febbraio 2004)
- [33] <http://www.appuntisoftware.it/ontology-web-language-owl-2/> - *Ontology Web Language (OWL)*. (10 ottobre 2010)
- [34] <http://www.foaf-project.org/> - *The Friend of a Friend (FOAF) project*.
- [35] <http://xmlns.com/foaf/spec/> - *FOAF Vocabulary Specification 0.98*. (Namespace Document 9 agosto 2010 - Marco Polo Edition)
- [36] <http://dublincore.org/documents/dces/> - *Dublin Core Metadata Element Set, Version 1.1*. (11 ottobre 2010)
- [37] <http://www.tutorialsonline.info/Common/DublinCore.html> - *Dublin Core Tutorial*.
- [38] <http://www.w3.org/2004/02/skos/> - *SKOS*.
- [39] <http://flora.sourceforge.net/aboutFlogic.php> - *F-logic*.
- [40] <http://www.w3.org/2002/ws/sawSDL/> - *Semantic Annotations for WSDL Working Group*.
- [41] <http://en.wikipedia.org/wiki/OWL-S> - *OWL-S*.
- [42] <http://www.w3.org/Submission/WSMO/> - *WSMO*.
- [43] <http://www.w3.org/Submission/WSML/> - *WSML*.
- [44] <http://ruleml.org/> *The Rule Markup Initiative*.
- [45] <http://www.w3.org/Submission/SWRL/> - *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. (W3C Member Submission 21 maggio 2004)
- [46] <https://oxygen.informatik.tu-cottbus.de/reverse-i1/?q=R2ML> - *R2ML - The REVERSE I1 Rule Markup Language*.
- [47] <http://www.w3.org/2005/rules/Overview.html> - *Rule Interchange Format*.

- [48] <http://www.w3.org/TR/rdf-sparql-query/> - *SPARQL Query Language for RDF*. (W3C Recommendation 15 gennaio 2008)
- [49] <http://www.openrdf.org/index.jspl> - *openRDF.org ...home of Sesame*.
- [50] <http://www.edutella.org/edutella.shtml> - *Edutella - P2P for the Semantic Web*.
- [51] <http://jena.sourceforge.net/> - *Jena - A Semantic Web Framework for Java*.
- [52] <http://triple.semanticweb.org/> - *TRIPLE*.
- [53] <http://kaon2.semanticweb.org/> - *KAON2*.
- [54] <http://flora.sourceforge.net/> - *FLORA-2: An Object-Oriented Knowledge Base Language*.
- [55] <http://microformats.org/> - *Microformats*.
- [56] <http://www.w3.org/TR/rdfa-syntax/> - *RDFa in XHTML: Syntax and Processing - A collection of attributes and processing rules for extending XHTML to support RDF*. (W3C Recommendation 14 ottobre 2008)
- [57] <http://en.wikipedia.org/wiki/RDFa> - *RDFa*.
- [58] <http://groupme.org> - *GroupMe!*.
- [59] <http://wwwis.win.tue.nl/hera/> - *The Hera Research Program*.
- [60] <http://www.personal-reader.de/wp/> - *Personal Reader Framework*.
- [61] <http://www.smartweb-project.org/> - *SmartWeb*.
- [62] <http://revyu.com/> - *Revyu*.
- [63] <http://code.google.com/intl/it-IT/apis/socialgraph/> - *Google Social Graph API*.
- [64] <http://chip-project.org/index.html> - *CHIP*.
- [65] <http://fire.rettorato.unito.it/blog/?id=42796> - *iCITY*.
- [66] <http://www.ubisworld.org/> - *UbisWorld*.
- [67] <http://echo.iat.sfu.ca/> - *ec(h)o*.
- [68] <http://trust.mindswap.org/FilmTrust/> - *FilmTrust*.
- [69] <http://bibster.semanticweb.org/> - *Bibster*.



- [70] [http://www.sis.pitt.edu/~paws/system\\_annotated.htm](http://www.sis.pitt.edu/~paws/system_annotated.htm) - *AnnotatEd*.
- [71] [http://www.sis.pitt.edu/~paws/system\\_knowledgesea2.htm](http://www.sis.pitt.edu/~paws/system_knowledgesea2.htm) - *Knowledge Sea II*.
- [72] <http://techcrunch.com/2005/06/29/profile-yahoo-my-web-20/> - *Profile: Yahoo My Web 2.0*.
- [73] <http://wordnet.princeton.edu/> - *WordNet - A lexical database for English*.
- [74] <http://dbellini.web.cs.unibo.it/> - *Pillo SAS's*.

# Ringraziamenti

Al termine di questo lavoro ci tengo a ringraziare:

- i miei amici e compagni di corso, con i quali sono cresciuto e ho studiato;
- la mia famiglia, che mi ha sostenuto e mi ha permesso di portare avanti i miei studi;
- il professor Mauro Gaspari, relatore di questa tesi, che mi ha aiutato nella realizzazione di questo lavoro.

Ma soprattutto tengo a ringraziare me stesso, per la grande pazienza e voglia di fare che posseggo da sempre, e che spero di non perdere mai.